A Continuum Mechanic Design Aid for Non-planar Compliant Mechanisms

by

Patrick Andreas Petri

S.B. Mechanical Engineering
Massachusetts Institute of Technology, 2001

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2002

Author .....
Department of Mechanical Engineering
August 9, 2002

Certified by .....
Martin L. Culpepper III
Assistant Professor of Mechanical Engineering
Thesis Supervisor

Accepted by .....
Ain A. Sonin
Chairman, Graduate Thesis Committee

A Continuum Mechanic Design Aid for Non-planar Compliant Mechanisms

by

Patrick Andreas Petri

Submitted to the Department of Mechanical Engineering
on August 9, 2002, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

This thesis documents the development of CoMeT, a conceptual evaluation and detailed synthesis aid for the design of compliant mechanisms. The vision behind CoMeT is making the limiting step in flexure design the speed of the user's imagination, not proficiency with software tools. Sophisticated kinematic analysis routines are seamlessly integrated into a three dimensional finite element program. A user may interface through both a convenient GUI and the powerful MATLAB command line.

CoMeT's element models have been shown to generally lie within 3% of traditional FEA predictions. The experimentally determined response of a typical complex mechanism differed by less than 10%, and CoMeT proved to be just as accurate as conventional FEA. In a brief user interaction study, a subject with one hour of CoMeT training was able to perform a two-variable optimization in half the time it took with traditional software. Observations suggest that CoMeT encourages the conceptual thought and high-level insights that are the key to success in mechanism design.

Thesis Supervisor: Martin L. Culpepper III
Title: Assistant Professor of Mechanical Engineering

Acknowledgments

I would like to dedicate this thesis to my parents, Ingrid and Wolfgang Petri, who have been an unfailing source of support, encouragement, and inspiration.

Next I want to thank my colleagues in the PSDAM lab, Gordon Anderson, Carlos Araque, Kelly Harper, Shih-Chi Chen, Naomi Davidson, Marcos Rodriguez, and Leeway Ho, for their company, advice, and support. Whenever I needed to share stipulations or vent concerns, you were there for me.

Professor Martin Culpepper has been the inceptor of this project and a relentless source of motivation. Where others see the physical limits of reality, he develops groundbreaking non-traditional mechanisms that become the subject of Master's theses, journal articles, and, someday, engineering folklore. Without him, PSDAM would never be the same.

MIT's Martin Fellowship and Professor Culpepper's TA and summer research funds have made this thesis financially possible, and I am grateful for the monetary independence I have gained in the process.

In closing, I would like to thank the MIT community for providing an environment that nurtures innovative projects, personal growth, and good-natured competition, while retaining an amiable social atmosphere.

Nomenclature

This is a list of principal variables used in this thesis. Some variable families only vary by subscript; in this case, the subscript is referred to explicitly and may be taken as variable. For example, when r_x is listed as a position vector along the x direction, it is understood that r_y must point in the y direction, and so forth. If a variable has context-sensitive meaning, such as β, the context follows that variable enclosed in square brackets. A few variables, such as the yield stress $sigma_y$, take on special meaning with particular subscripts; their definitions do not make an explicit reference to the subscript letter.

All scalar variables are in italics. Boldface denotes matrices and vectors. Expressions in regular typeface are abbreviations. Typewriter typeface is used for all MATLAB expressions.

Subscripts

A	Actuator
$\mathcal{A}$	In coordinate frame tangent to curved beam. See Fig. 2-12.
B	Backward
f	Final (depends on context)
F	Forward
$\mathbf{g}$	Expressed in the global frame, pertaining to a single element
$\mathbf{G}$	Expressed in the global frame, pertaining to the entire structure
i	Pertaining to a specific element, but true for comparable elements
j	Similar to i, but used where i is already in use
$\mathbf{k}$	Known
ℓ	Expressed in the local (element) coordinate frame
o (naught)	Original (context dependent)
$\mathbf{u}$	Unknown

Subscripts, cont.

r	Radial
t	Tangential
x	In the x direction
x	Based on zero displacement tests, see Subsec. 3.2.1
y	In the y direction
z	In the z direction
0 (zero)	Many meanings, defined in context

Abbreviations

CoMeT	Compliant Mechanisms Tool
DOF	Degrees of Freedom
FEA	Finite Element Analysis
GUI	Graphical User Interface
MEMS	Micro Electro-Mechanical Systems
MIT	Massachusetts Institute of Technology
PSDAM	Precision Design and Manufacturing lab
SPD	Symmetric Positive Definite

Roman Variables

a [most]	The larger of the half-width and half-height of a beam cross section, see Eq. (2.39)
a [curved]	Curved beam equivalent of w, see Fig. 2-11
a_o, a_f	Tapered beam width. See Fig. 2-10
A	Cross-sectional area of a beam element

Roman Variables, cont.

b [most]	The smaller of the half-width and half-height of a beam cross section, see Eq. (2.39)
b [curved]	Curved beam equivalent of h, see Fig. 2-11
$b_o,\ b_f$	Tapered beam height. See Fig. 2-10
B	The number of beams in a structure
c	Pertains to curved beam curvature: $c \equiv \cos\theta$
C	Centroid point
C_i	(i an integer) Temporary variables used to construct $\zeta_{\mathcal{A}}$
$\mathbf{C}$	Compliance matrix
$\mathbf{C_G}$	The global compliance matrix
C_{F_x}, etc.	Tapered beam (linear) compliance due to F_x. See Eq. (2.64)
$C_{\mathrm{x}F_x}$, etc.	Curved beam x-compliance due to F_x. See Eqns. (2.90)-(2.93)
$\mathbf{C}(\mathbf{r})$	Matrix that translates vectors by $\mathbf{r}$
E	Young's modulus
F	Scalar force
$\mathbf{F}$	Generalized force, including forces and moments
$\mathbf{F_{G_o}}$	Generalized force acting on entire structure, including grounded nodes. See Eq. (2.10)
G [most]	Shear modulus
G [η-T]	Force transfer function, see Eq. (3.14)
`Gr`	The number of grounded nodes in a structure
h	Height of a beam element (along the local z direction)
H	Displacement transfer function, see Eq. (3.7)
$\widehat{\ }$ (hat)	(overscript) Vector of unit magnitude
$I_x,\ I_y$	Area moment of inertia about the x- and y-axes, respectively
IC	Instant center point
J	Torsional moment of inertia of a beam element cross section
K	Scalar stiffness

Roman Variables, cont.

$\mathbf{K}$	Stiffness matrix
$\mathbf{K_G}$	The invertible global stiffness matrix, after grounded node entries are truncated
$\mathbf{K_{G_o}}$	The full, singular global stiffness matrix
l	Length of a beam element, as defined in Eq. (2.22)
L	Same as l
M	Scalar moment
N	A given node *or* the number of nodes in a structure
NL	The number of loads on a structure
$\mathbf{r}$	Generic position vector
r_x	Scalar component of generic position vector along x-direction
Δr_x	x-coordinate offsets, see Eq. (2.21)
R	Radius of a curved beam
$\mathbf{R}_P$	Position vector (coordinates) of point P in the global frame
$\mathbf{R_x}$	Matrix that rotates quantities around the current x-axis
$\mathbf{R}_{P-Q}$	Vector pointing from point P to point Q
$\mathbf{R}_{0/1}$	[3x3] Rotation matrix that relates frame 1 to base frame 0
s	Pertains to curved beam curvature: $s \equiv \sin\theta$
$^{\mathrm{T}}$	(exponent) Matrix transpose
$^{-\mathrm{T}}$	(exponent) Inverse matrix transpose. Note that $\left(\mathbf{A}^{\mathrm{T}}\right)^{-1} = \left(\mathbf{A}^{-1}\right)^{\mathrm{T}}$
T	The true, invariant transmission ratio, see Eq. (3.10)
T_F	T, derived using force tests. See Sec 3.2.1
T_x	T, derived using displacement tests. See Sec 3.2.1
$\mathbf{T}_{\mathrm{i/j}}$	A transformation matrix that relates frame j to frame i
$\mathbf{T}_l$	l-offset transformation. See Eq. (2.33)
w	Width of a beam element (along the local y direction)
W	Energy

Roman Variables, cont.

W_{strain}	Strain energy. See Eq. (3.32)
$\mathbf{x}$	Generalized displacement, including linear displacements and rotations
x	Base of an xyz coordinate system pointing rightward on paper
X	Generic parameter for sensitivity study
$\mathbf{x_{G_o}}$	The generalized displacement vector of all nodes in a structure, including grounded nodes. See Eq. (2.10)
y	Base of an xyz coordinate system pointing upward on paper
z	Base of an xyz coordinate system pointing out of the paper
Z	Generic criterion of sensitivity study

Greek Variables

α [tapered]	Tapered beam width attenuation factor. See Eq. (2.56)
α [round]	Round beam planar moment stress factor. See Eq. (2.140)
α [curved]	Angular parameter giving the location of a curved beam cross section, measured from the apex of the arc
β [tapered]	Tapered beam height attenuation factor. See Eq. (2.56)
β [round]	Round beam non-planar moment stress factor. See Eq. (2.140)
β [curved]	Curved beam twist-to-bend ratio parameter
δ_P	Linear displacement vector of point P
ϵ	Small dimensionless perturbation parameter used in sensitivity study, see Eq. (3.34)
η	The transmission loss factor, see Eq. (3.10)
γ	Small perturbation length parameter in sensitivity study, see Eq. (3.35)
λ	Geometric scaling factor, see Sec. 4.3
ρ	Density of a structure's material

Greek Variables, cont.

σ	Stress
σ_y	Yield stress
τ	Shear stress
θ	One half the subtended angle of a curved beam
θ_x	Angle between coordinate systems, viewed perpendicular to the x-axis
θ_P	Angular displacement (rotation) vector of point P
$\zeta_{\mathcal{A}}$	Curved beam compliance matrix for non-planar forcing, expressed in a frame tangent to the beam, see Eq. (2.116)
ζ_{F_x}, etc.	Beam angular compliance due to F_x, see Eqns. (2.64) and (2.92)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis documents the analytic development of CoMeT, or Compliant Mechanism Tool. CoMeT is a conceptual evaluation and detailed synthesis aid for the design of non-traditional compliant mechanisms. Featuring sophisticated kinematic analysis routines integrated into a three dimensional finite element program, CoMeT greatly accelerates the product development cycle. Students of continuum mechanics can also benefit from CoMeT, since the MATLAB program is designed to help the user quickly build and verify an intuition for elastic structures. Structural engineers from various fields should appreciate the convenience of CoMeT's unique three dimensional sketching tool, instant preview, and rigorous analysis modules.

In this chapter, the fundamental issues that led to CoMeT's development are introduced. Next, the hypothesis is presented and the means of its evaluation covered. The chapter ends with an overview of the program structure and operation.

1.1 Motivation

1.1.1 Flexure Advantages

Wherever two parts of a mechanism move relative to each other, constraints must exist to keep them in proximity, control their orientation, and guide their motion. The devices commonly used can be classified into four distinct categories, depending on their mode of contact, as listed in Table 1.1.

Table 1.1: **The four major classes of motion constraints**

Contact Mode	Examples
Sliding	Bushings, lead screws, slideways
Rolling	Ball bearings, needle bearings, pulleys
Fluid	Air, water, and oil-based bearings
Quasi-rigid	Flexures

While each class in Table 1.1 has its strengths and weaknesses, flexures have many unique characteristics that make them the only viable candidate in many applications. Flexures rely on elastic deformations, coupled with differences in directional compliance, to induce motion constraints. Among their advantages are backlash-free motion, micro-scalability, single part construction, quiet operation, no need for lubrication, and immunity to contamination. These advantages are discussed in depth below, but first the relationship between flexures and compliant mechanisms warrants elucidation.

Compliant mechanisms may be defined as mechanisms that incorporate one or more flexural elements. When combined into more complex configurations, flexures retain their fundamental advantages, and the compliant mechanism concept inherits them. Since compliant mechanisms may incorporate many other features and are too diverse to discuss at length, it is difficult to generalize their usefulness. However, everything to follow in this section applies to compliant mechanisms as much as it

does to the flexures of which they are composed.

Flexures are the only motion constraint inherently free from backlash and other hysteresis, provided they operate outside the regions where creep and plasticity come into play. While bearings can be pre-loaded to minimize backlash, imperfections always remain, and costs rise rapidly as tolerances are tightened. A pre-loaded bearing or slideway actually employs a hybrid contact mode, since it utilizes a material's elastic properties, and performs better than its rigid counterpart. However, contact surface irregularities still limit the precision of pre-loaded mechanisms.

Secondly, flexures can be scaled much more easily than their alternatives. This is strongly coupled to the advantage that they permit monolithic construction. In nano and mesoscale design, assembly is often the greatest obstacle to success, since the devices to be built tend to be smaller than the tools that make them. Self-assembly is one promising approach, but nothing can match the simplicity of no assembly requirements in the first place. In addition, many useful flexures can be built with only planar features, which are well suited for lithography and other microfabrication techniques. Although CoMeT is fully functional in three dimensions, it was coded with the expectation that a substantial portion of applications would be planar or laminar. Care was taken to avoid three dimensional complications where two dimensional functionality would suffice.

Since flexures do not involve sliding or rolling contact, they operate quietly. Noise is not only a cause of consumer discomfort, it often has real functional implications. Often it is associated with energy dissipation, which in turn implies some degree of wear and tear. Energy often propagates through a mechanism via shock waves, which may disturb a precision device, especially if resonance is induced. Military applications often rely on stealth, and noise that is imperceptible to a human may well exceed the threshold of an enemy's tracking device.

Flexures, assuming they are constructed of a stable material, do not rely or stand to benefit from lubrication. Almost all other motion constraints require lubrication of some form. Fluid-contact elements may not need grease, but managing, sealing, and handling the fluid itself is usually more difficult than keeping parts lubricated.

Those applications that can function without lubrication usually do so at the expense of product durability and energy efficiency.

As a final argument, flexures are superior to other motion constraints in that they are essentially immune to contamination. While great care must be taken that bushings and bearings not be degraded by abrasive particles, flexures can operate in any environment that does not attack their constituent material. Unless macroscopic particles grossly impede their motion, flexures are insensitive to material disturbances. In practical systems ranging from MEMS to machinery, contamination is almost inevitable, and precision suffers accordingly.

1.1.2 Flexure Drawbacks

Naturally, flexures also have drawbacks. Range of motion is limited, disturbance forces may be hard to counteract, substantial amounts of energy must usually stored in the structure, strain hardening and creep may change system characteristics, and parasitic motions may lead to inaccuracies. However, the largest obstacle to flexure design seems to be the difficulty associated with their synthesis and analysis. According to Barber's Intermediate Mechanics of Materials [7], published in 2001,

> ... a finite element calculation of the stresses in a fairly simple three-dimentional component might involve two-days work for an experienced analyst and cost $2000, including salary, computing services and overhead costs. At this rate, the stress analysis alone for a fairly modest device might cost around $50,000. In addition, we shall probably need to perform other design calculations ...

Compliant Mechanisms lend themselves well to allowing multiple degrees of freedom, but these are usually moderately to highly coupled, making it difficult to isolate modules that can be optimized individually. A typical compliant positioning stage requires knowledge of the quantities listed in Table 1.2.

All of these objectives are coupled, even for a single degree of freedom device. In addition, a successful design will involve conceptual iterations and parameter

Table 1.2: **Quantities of interest in a compliant positioning stage**

• Maximum stress under several loading cases
• Work volume
• Actuator input stiffness
• Actuator-centroid control transfer function
• Sensitivity to disturbances
• Sensitivity to manufacturing imperfections
• Total volume and mass

optimizations. If this process is not partially automated, the costs involved may be prohibitive to the economic feasibility of the device under consideration. As a result, the field of compliant mechanisms is a vastly unexplored territory.

1.1.3 CoMeT's Role in Compliant Mechanism Design

The vision behind CoMeT is to make flexure design and analysis so easy, fast, and convenient that a minimally trained user can enjoy the fruits of his creativity the majority of the time, while spending very little effort clicking dialog boxes, customizing properties, and tending to details. The limiting step in flexure design should be the speed of the user's imagination, not the user's proficiency with software tools. A computer cannot replace the creative facilities of a human, but it can handle the mechanical tasks at hand, provide immediate intuition validation, and unleash the human's associative thinking ability.

CoMeT performs exactly what a compliant mechanism designer needs by combining a host of known concepts, and a few new ones, into one convenient package. The only allowed elements are rigid plates and several common types of beams. A civil engineer would deem them frame elements, since they allow tensional compliance and torsional resistance, effects that are neglected for a prototypical beam element. However, the author, being a mechanical engineer, believes that the word "frame" implies architectural applications, which is not the primary focus of CoMeT. In fairness to both disciplines, the two-node elements in CoMeT will primarily be referred to simply as "elements."

By forcing the designer to represent structures as a combination of rigid and two-node elements, it encourages good design habits, since details such as rounds, holes, and fillets can usually be added later. The user is forced to face the fundamental design choices that appear at the early stages of a design cycle. In addition, it permits the program to compute the quantities listed in Table 1.2 with improved speed. In the very end of the design phase, when all conceptual decisions and major parameter optimizations have been performed, a traditional finite element analysis can be run once to re-gain what is usually a few percent of lost accuracy.

Current programs cannot match CoMeT's efficient analysis package. Traditional finite element analysis programs offer complete three-dimensional solutions to beam element configurations [8], but provide so many functions that they are difficult to learn, slow to use, and expensive [2]. Until recently, a finite element analyst had to study the subject for several weeks, learning how to write code to extract the information he or she needs, as well as tricks that reduce computation time, mesh spacing inaccuracies, and singularities. Market demand and increasing computational availability have brought about several CAD-integrated FEA packages, such as CosmosWorks for SolidWorks and Pro/Mechanica for Pro/Engineer. While such programs are vastly easier to learn than previous software, and code need no longer requires manual editing, the other objections still apply. Using elements other than the default requires considerable expertise, and licenses cost several thousands of dollars.

Civil Engineers make extensive use of beam and frame elements, since breaking an entire building into a customary solid FEA mesh would be impractical. The immense market for building analysis has spawned over one hundred frame-based FEA packages [1]. Although these programs may be excellent tools in their own right, they are all highly specialized. For example, one program [6] allows loads to be defined as either "snow" or "non-snow." While this may save a civil engineer the plight of thumbing through volumes of building codes, it loses relevance for compliant mechanism designers.

As a result of software being either too broad or specialized for other applications,

compliant mechanism designers have been forced to rely on their intuition. Most useful flexure concepts known to date are either four-bar mechanisms or conglomerations of parallel beam elements, as these can be synthesized from analogies to rigid linkages. Even these are seldom particularly optimized, with order-of-magnitude hand calculations taking the place of formal studies. On top of that, it often takes an engineer with years of experience to suggest conceptual improvements, since an intuition is only built through arduous trial and error.

In an effort to counteract the need for experience and human fallibility, several computational tools have been created that synthesize flexures [4, 12, 15]. Although future improvements are likely, these software tools have significant limitations. First, all current approaches only consider planar structures. Second, optimization algorithms are usually genetic, and "mutate" gradually from a seed scenario toward a desired, stable configuration, should they converge. This means that radically new solutions can only be found with good fortune, not through systematic derivation. The main and most limiting factor of computer-driven flexure synthesis is that solutions can only be obtained after the underlying problem is fully formulated. Human creativity can progress with only a partial understanding of the problem, readily recognize patterns, and generate "educated guesses". A vague solution can gradually be narrowed until it becomes concrete. It would therefore be more effective to design flexures directly than to design an algorithm to design them. Even if a whole family of related flexures required optimization, it would be more efficient and reliable to understand the pattern, perhaps by deriving a governing equation, than to search until it appeared that no further improvements were possible.

1.2 Background

The Hexflex^{TM}, pictured in Fig. 1-1, is a unique compliant spatial mechanism developed in the Precision Systems Design and Manufacturing lab (PSDAM) at MIT in 2001. Analytic models required many days to develop and solve, yet continued

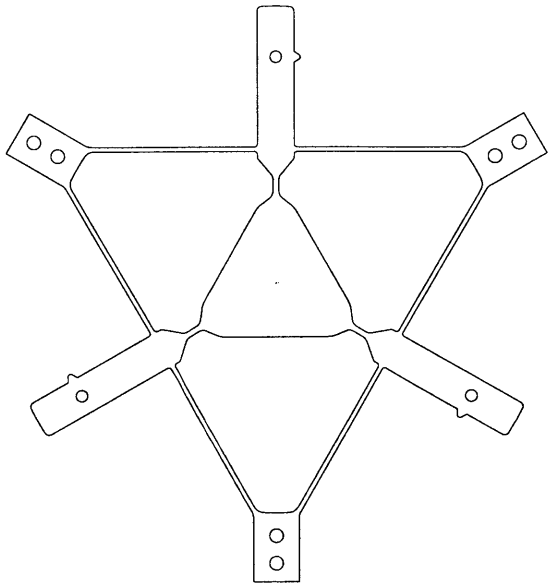


Figure 1-1: The Hexflex^{TM} positioning system

to elude sufficiently precise correlation with experimental data. Many non-intuitive effects not easily captured in simplified beam models govern the Hexflex behavior. Traditional finite element analysis yielded accurate deflections and stresses, but the remaining quantities in Table 1.2 could only be obtained through dozens of slow iterations and complex analytical work. In addition, proposed conceptual design changes demanded repeated application of the entire procedure.

It is at this point that the CoMeT concept was conceived. The seed vision was to create a tool which could be used by designers for rapid first order concept design of compliant mechanisms. The tool would be designed to integrate the experience of a designer with a powerful analysis program, thereby enabling rapid generation of novel compliant mechanisms. This marriage of experience and analytic power would occur via a GUI that allows the user to create, evaluate, modify, and re-evaluate a design in less than two minutes and with better than 10% accuracy of performance.

Although it may be appear more difficult to solve the general linear elasticity problem than to solve a specific problem, this is not entirely true. Large pieces of the work had been done before, just not as part of one integrated whole. In addition, generalized formulations may be more difficult to grasp conceptually, but

are usually much easier to implement, since the same procedure is repeated many times. Moreover, given the list of proposed conceptual changes to the Hexflex, and the promise of other compliant mechanisms to be designed in the future, the point of justified return would surely be reached. Finally, it was determined that additional benefits would be gained from employing CoMeT as a teaching tool.

Once developed, CoMeT should have considerable impact on precision engineering. MEMS, optical technology, and semiconductor manufacturing all require high precision. As discussed in Section 1.1, flexures have many intrinsic advantages that suggest their use in precision applications, but are usually difficult to analyze. If CoMeT can break the analysis barrier, help engineers acquire an intuition for compliant mechanisms, empower their creativity, and shorten the product development cycle, it can help us develop these emerging technologies faster. Put simply, CoMeT is an enabling technology for next generation technologies.

1.3 Hypothesis

The hypothesis examined in this thesis is that a program can be developed whose predictions lie within 10% of traditional FEA programs, but can be used to design flexures more effectively and in less time than it would take with current tools. Furthermore, it is worth noting whether an actual physical system matches the analytic predictions of CoMeT and traditional programs. Although hypothetical discrepancies are likely to be a function of the application, it is nonetheless instructive to analyze CoMeT in an applied environment.

The metrics we will use to evaluate success are as follows:

- The accuracy of CoMeT formulas is assessed via comparisons of CoMeT to CosmosWorks® calculations. The behavior of both single elements and a complex system, namely the Hexflex™, will be scrutinized. No compliance or stress should differ by more than 10%.

- Time will be used as a quantitative measure to demonstrate that CoMeT reduces

development time.

- Physical flexural hardware compliance should not differ by more than 15% from its CoMeT idealization.

1.4 Scope

Although the CoMeT concept will be highly versatile, this thesis must have finite scope. Other features may be added later, as it becomes clearer what functionality is most useful. The key to allowing retro-fits is making CoMeT sufficiently modular and adaptable. This requires well-planned programming schemes and an effort to avoid limitations that may fatally limit CoMeT in the future.

In this thesis, the continuum mechanic and kinematic analyses will be limited to the isotropic and linear regime. Anisotropic materials are fairly rare in flexure design, and the most common, Silicon, is fairly stiff and brittle. Linear means that all deflections, whether linear or angular, must be directly proportional to forces and moments on each element. Geometric nonlinearity, contact mechanics, buckling, dynamics, Coulomb friction, plasticity, creep, and strain hardening analyses are therefore excluded. I have made this decision because 95% of current compliant mechanisms can be understood to first order through linear analysis. Once a system becomes nonlinear, the existence of solutions is no longer guaranteed, responses may be path dependent, superposition no longer holds, and matrix inversions must be augmented with iterative methods. Many FEA packages exist that perform all the aforementioned calculations, and if a design is evolved to the point where second order effects demand attention, commercial software can be used.

With respect to kinematic constraints, a given node must be either completely grounded (constrained to zero linear and angular displacements) or free to satisfy force equilibrium. This decision is based on the consideration that most compliant mechanisms would become inaccurate if they included sliding or rubbing interfaces. That said, it is still possible to obtain deformation responses in CoMeT when input

displacements are known. The procedure is explained in Sec. 2.7. Alternatively, cantilever beams may be used to approximate partial constraints.

Emphasis will remain on post analysis that fits the aforementioned framework, as well as development of an intuitive, user-friendly, convenient graphical user interface (GUI). Thereby, large rewards may be reaped from ordinary effort. Without an appropriate interface, it is difficult to see CoMeT gain widespread acceptance. Without post analysis modules, CoMeT would differ little from existing frame-based FEA programs. Chapter 3 presents the post processing modules, while Chapter 4 covers the GUI in detail.

1.5 Program Structure Overview

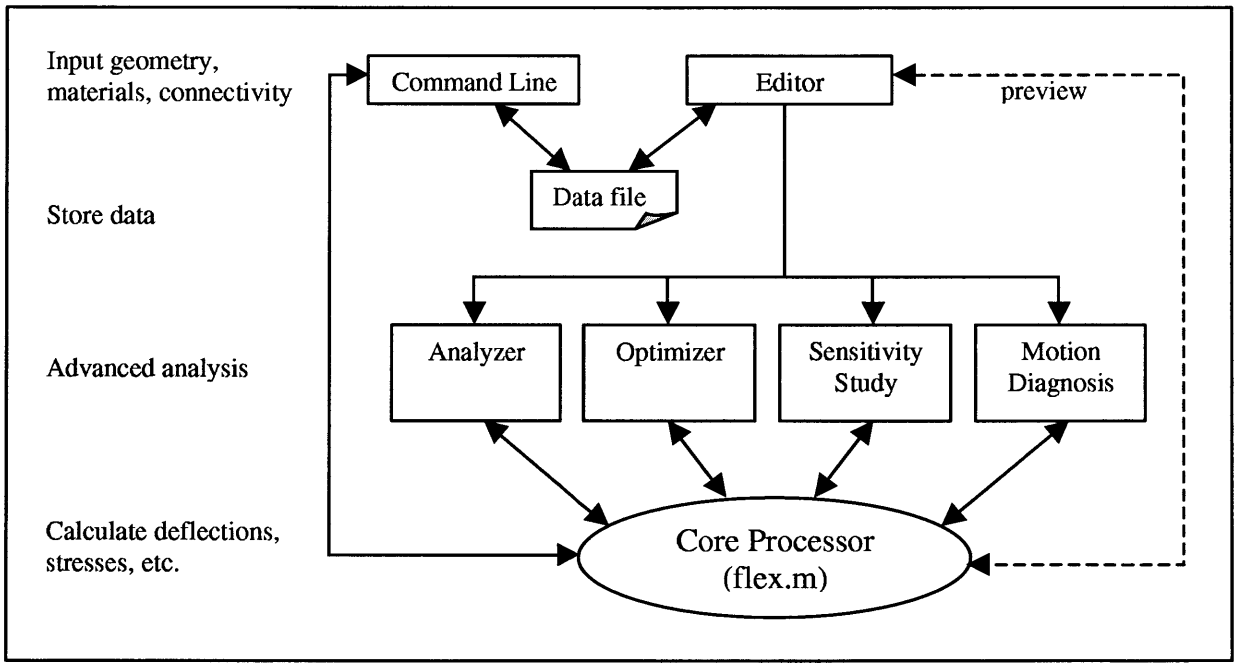


Figure 1-2: Program structure overview

The basic structure of CoMeT is shown in Fig. 1-2. Arrows indicate information flow, usually in the form of function calls and data retrieval, while each box represents a CoMeT module. The "core processor", indicated by an oval, corresponds directly to

the function `flex.m`. Most of Chapter 2 is devoted to the workings of `flex.m`, written on a general level that requires no knowledge of programming. Chapter 4 elaborates on some of the subtleties required to implement the concepts from Chapter 2 in code.

Graphical user interfaces (GUIs) are the *de facto* standard in major modern programs, and for good reason. They offer effortless data entry, visual error recognition, intuitive interaction patterns, and æsthetic appeal. This is a natural consequence of humans' visual processing capabilities. However, GUIs can never match the flexibility of a command line interface in all regards. While common functions can usually be accelerated in a graphical environment, many more sophisticated tasks are simply best suited for a keyboard. A GUI could incorporate a host of complex functions, but would lose the benefit of simplicity. It would also tend to become so "cluttered" that the accessibility of common tools would invariably suffer.

In order to capture both the visual power of a GUI and the versatility of a command line, CoMeT is designed to interact through both. This is unusual in mainstream applications, but CoMeT is intended for academic and research environments, where demands are seemingly unsatiable. The harmonious coexistence of two such dissimilar interfaces is only possible through a strict data file protocol. However strict, the format requirements are nonetheless simple and intuitive.

As shown in Fig. 1-2, both the GUI editor and the command line may read and write to this common data file. The interfaces may take arbitrary turns modifying the data. The file contains all the information necessary to define a structure study — geometry, material properties, connectivity, loading conditions, motion constraints, and points of special interest henceforth known as centroids. The command line may call the core processor directly, receiving the system deformation response, stress distribution, and other information. Thereby, data interfaces seamlessly into other MATLAB® toolboxes, custom scripts of the user's choosing, or, in textual form, foreign software (such as Microsoft® Excel™, PowerPoint™, or Labview™).

The GUI editor may also call the core processor when in instant preview mode, but usually will pass the data file information directly to an advanced analysis

module. These modules include the analyzer, the optimizer, a sensitivity study, and a motion diagnosis tool. Each calls the core processor repeatedly, capitalizing on the computational speed of `flex.m`. Advanced modules are discussed in Chapter 3.

Chapter 2

Continuum Mechanics and Kinematics

The strictly linear analysis in this thesis is based on the direct stiffness approach described by Gurley [13] and others [8, 9]. Energy methods and other approaches may prove valuable when nonlinearities are considered, but remain inherently more complicated for the task at hand. Castiglione's Theorem, for example, requires partial differentiation, which is difficult to implement in code. Instead, a system of coupled linear equations, embodied in the global stiffness matrix $\mathbf{K_G}$, can be constructed systematically, and solving them is a simple matter of calling a built-in matrix inversion routine.

There are several more computationally efficient alternatives to explicitly inverting the global stiffness matrix, such as left division or wavefront propagation [8]. However, since a typical compliant mechanism of medium complexity may contain on the order of 100 degrees of freedom (DOF), inversion is not nearly as problematic as it would be for a finely meshed finite element model, where the number of DOF is typically three orders of magnitude higher. In addition, the possibility of defining rigid plates further reduces the DOF in $\mathbf{K_G}$. When inverted, $\mathbf{K_G}$ becomes the global compliance matrix $\mathbf{C_G}$. It contains all the information necessary to compute deflections of every

node for *any* nodal load set. Clearly this is inappropriate for common FEA programs, which compute the response to one loading case at a time. In summary, computing $\mathbf{C_G}$ is equivalent to finding every possible forcing response in one operation.

MATLAB inverts quite efficiently by taking advantage of the fact that $\mathbf{K_G}$ is sparse and symmetric positive definite (SPD). In practice, inversion of a typical compliant mechanism takes less than 3ms on a 1.9GHz computer. The entire analysis of the same structure took 60ms (UPDATE WHEN DONE!), which is faster than a typical human reflex. As Moore's law continues to govern the progress of computing power, the computational delay of solving the deformation response of a structure should become unnoticeable in the immediate future.

2.1 Direct Stiffness Approach

As long as deflections are small and material does not yield, as discussed in Sec. 1.4, finite elements can be represented by some linear stiffness matrix:

$$\mathbf{F}_\ell = \mathbf{K}_\ell \, \mathbf{x}_\ell \tag{2.1}$$

The subscript ℓ denotes a frame local to the element. Subsequently, g will signify quantities in the global frame pertaining to a given element, and G will represent global quantities for the entire structure. Note that $\mathbf{K}_\ell$ must be SPD [14], as dictated by the principle of reciprocal forces and the first law of thermodynamics.

The transformation matrix $\mathbf{T}_{0/1}$, covered in detail in Sec. 2.2, serves as a bridge between local and global frames. In Eq. (2.15), $\mathbf{T}_{0/1}$ is a [6x6] matrix, but in this context the size of $\mathbf{T}_{0/1}$ would reflect the degrees of freedom of the element it was transforming (the elements used in CoMeT, with the exception of rigid plates, each have 12 DOF). Local and global generalized displacements — vectors containing linear

and angular deflections of an element's nodes — are related as follows:

$$\mathbf{x}_\ell = \mathbf{T}_{0/1}^{\mathrm{T}}\,\mathbf{x}_\mathbf{g} \tag{2.2}$$

$$\mathbf{x}_\mathbf{g} = \mathbf{T}_{0/1}^{-\mathrm{T}},\mathbf{x}_\ell \tag{2.3}$$

Generalized forces — vectors combining forces and moments at element nodes — obey similar relations:

$$\mathbf{F}_\ell = \mathbf{T}_{0/1}^{-\mathrm{T}}\,\mathbf{F}_\mathbf{g} \tag{2.4}$$

$$\mathbf{F}_\mathbf{g} = \mathbf{T}_{0/1}\,\mathbf{F}_\ell \tag{2.5}$$

We can combine Eqns (2.1), (2.2), and (2.4) to obtain

$$\mathbf{F}_\mathbf{g} = \mathbf{T}_{0/1}\mathbf{K}_\ell\,\mathbf{T}_{0/1}^{\mathrm{T}}\,\mathbf{x}_\mathbf{g}\,. \tag{2.6}$$

If we define

$$\mathbf{K}_\mathbf{g} = \mathbf{T}_{0/1}\mathbf{K}_\ell\,\mathbf{T}_{0/1}^{\mathrm{T}}\,, \tag{2.7}$$

Eq. (2.6) becomes

$$\mathbf{F}_\mathbf{g} = \mathbf{K}_\mathbf{g}\,\mathbf{x}_\mathbf{g}\,. \tag{2.8}$$

$\mathbf{K}_\mathbf{g}$ will also be SPD, since $\mathbf{T}_{0/1}$ is purely real (i.e. it has no imaginary or complex components). Next, we assemble the global structural stiffness matrix $\mathbf{K}_{\mathbf{G_o}}$ from element stiffness matrices $\mathbf{K}_{\mathbf{g},i}$ by summing over all beams,

$$\mathbf{K}_{\mathbf{G_o}} = \sum_{i=1}^{B} \mathbf{K}_{\mathbf{g},i}\,, \tag{2.9}$$

where $\mathbf{K}_{\mathbf{g},i}$ must be written into the appropriate fields of $\mathbf{K}_{\mathbf{G_o}}$. If beam i connects node j to node k, fields jj, jk, kj, and kk would be affected, as shown in Fig. 2-1.

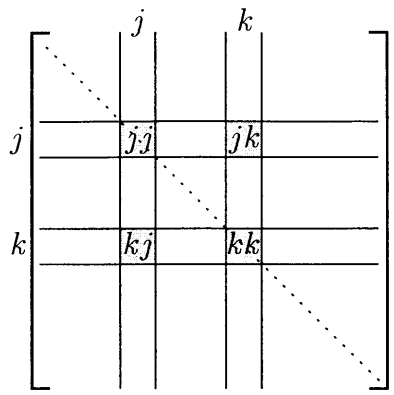


Figure 2-1: Regions of $\mathbf{K_{G_o}}$ affected by addition of $\mathbf{K_{g,i}}$

Since each addition is symmetrically placed, $\mathbf{K_{G_o}}$ is also SPD. We can now say

$$\mathbf{F_{G_o}} = \mathbf{K_{G_o}}\,\mathbf{x_{G_o}}\,. \tag{2.10}$$

Every node in the structure is either grounded or free. Grounded nodes have zero generalized displacements, and the loads acting on free nodes must be specified. (Since the majority of nodes will not be loaded, most applied forces and moments will be set to zero, which constitutes an easily overlooked constraint.) Therefore, we have a complete set of equations that can be solved for all unknown quantities.

If grounded nodes are listed last, Eq. (2.10) can be written in block form, k denoting known quantities, u unknown:

$$\begin{bmatrix} \mathbf{F_k} \\ \mathbf{F_u} \end{bmatrix} = \begin{bmatrix} \mathbf{K_{ku}} & \mathbf{K_{k0}} \\ \mathbf{K_{uu}} & \mathbf{K_{u0}} \end{bmatrix} \begin{bmatrix} \mathbf{x_u} \\ \mathbf{0} \end{bmatrix} \tag{2.11}$$

For now, only $\mathbf{K_{ku}}$ matters. In fact, although the element-level components of $\mathbf{K_{uu}}$ are necessary for force recovery (section 2.3), $\mathbf{K_{uu}}$ need not be constructed. Considering Eq. (2.12) a definition for Eq. (2.13), we can parallel the elegant notation of Eq. (2.6):

$$\mathbf{F_k} \;=\; \mathbf{K_{ku}}\,\mathbf{x_u} \tag{2.12}$$

$$\mathbf{F_G} = \mathbf{K_G}\,\mathbf{x_G} \tag{2.13}$$

In the program, the summation in Eq. (2.9) simply ignores entries pertaining to grounded nodes, and $\mathbf{K_G}$ is created directly.

Since $\mathbf{K_G}$ is SPD, its inverse, the global compliance matrix $\mathbf{C_G}$, must exist.

$$\mathbf{x_G} = \mathbf{C_G}\,\mathbf{F_G} \tag{2.14}$$

Thus, the quasi-static deformation response of any loaded structure can be found.

2.2 Transformation Matrices

Transformation matrices are an integral component of CoMeT. Hale and Slocum [14] present an excellent explanation of what they are and how they are constructed. Transformation matrices are best understood as entities that transform vectors from a local frame to a base frame. If the base frame is numbered 0, and the local frame given the label 1, the subscript notation in $\mathbf{T}_{0/1}$ suggests which frame should properly be written on either side of the matrix. This can be seen in Eq. (2.5) and more conspicuously in later equations such as (2.128), where internal subscripts tend to "cancel" as in Einstein notation. As to the mechanics of constructing $\mathbf{T}_{0/1}$, a summary of Hale and Slocum's explanation follows.

$$\mathbf{T}_{0/1} = \begin{bmatrix} \mathbf{R}_{0/1} & \mathbf{0} \\ \mathbf{C}\left(\mathbf{r}_0\right)\mathbf{R}_{0/1} & \mathbf{R}_{0/1} \end{bmatrix} \tag{2.15}$$

$\mathbf{C}\left(\mathbf{r}_0\right)$ is just a cross product matrix of the vector $\mathbf{r}_0$, which points from the origin of 0 to that of 1. Breaking $\mathbf{r}$ into (global) components and letting it act upon an

arbitrary vector $\mathbf{a}$,

$$\mathbf{r} \times \mathbf{a} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \times \mathbf{a} = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \mathbf{a} = \mathbf{C}(\mathbf{r})\, \mathbf{a} \tag{2.16}$$

$\mathbf{R}_{0/1}$ is a standard [3x3] rotation matrix that accounts for the difference in angular orientation between coordinate systems 0 and 1. There are many ways to define a rotation matrix; here, the xyz-convention is used.

$$\mathbf{R}_{0/1} = \mathbf{R}_z(\theta_z)\, \mathbf{R}_y(\theta_y)\, \mathbf{R}_x(\theta_x) \tag{2.17}$$

$$\mathbf{R}_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \tag{2.18}$$

$$\mathbf{R}_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{2.19}$$

$$\mathbf{R}_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.20}$$

One advantage of the xyz-convention is that if θ_x, the angular orientation of an element throughout its volume, is zero, the top face of the element will continue to appear as a line when projected to the global xy-plane. For example, a structure extruded from an outline in the xy-plane would have all θ_x set to zero. For more complex structures, the user need only determine how far beams are rotated beyond this default orientation. The xyz-convention minimizes the inherent difficulty of three dimensional visualization by decoupling θ_x rotations from the remaining transformations.

Figure 2-2 illustrates how the local and global xz planes of a rectangular beam

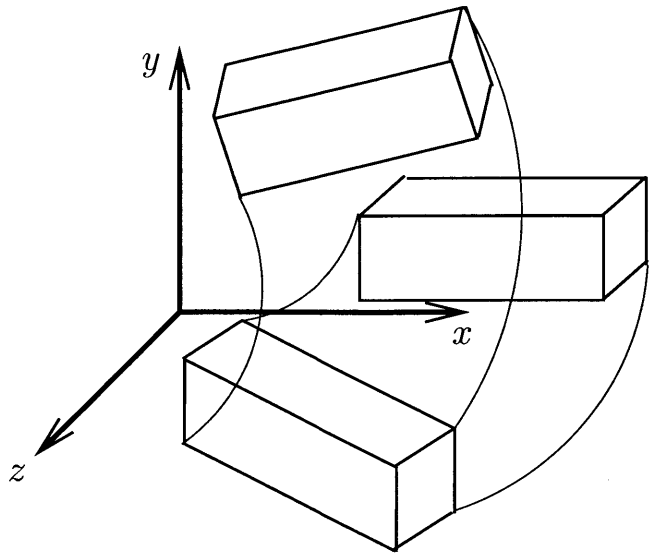


Figure 2-2: Beam with $\theta_x = 0$ transformed by $\mathbf{T}_{0/1}$

remain parallel when $\theta_x = 0$.

While the user must specify θ_x, element node coordinates determine l, θ_y, and θ_z. Let Δr represent coordinate offsets. In this context, 1 and 2 refer to the left and right nodes of an element, respectively.

$$\begin{bmatrix} \Delta r_x \\ \Delta r_y \\ \Delta r_z \end{bmatrix} \equiv \begin{bmatrix} r_{x,2} \\ r_{y,2} \\ r_{z,2} \end{bmatrix} - \begin{bmatrix} r_{x,1} \\ r_{y,1} \\ r_{z,1} \end{bmatrix} \tag{2.21}$$

The length of the beam can be determined from the three dimensional pythagorean theorem.

$$l = \sqrt{{\Delta r_x}^2 + {\Delta r_y}^2 + {\Delta r_z}^2}\,. \tag{2.22}$$

The θ_z relation, illustrated in Fig. 2-3, is

$$\theta_z = \text{atan2}(\Delta r_x,\ \Delta r_y) \tag{2.23}$$

Here 'atan2' can be considered MATLAB syntax. The `atan2` command finds the polar angle of (x, y), retaining quadrant information.

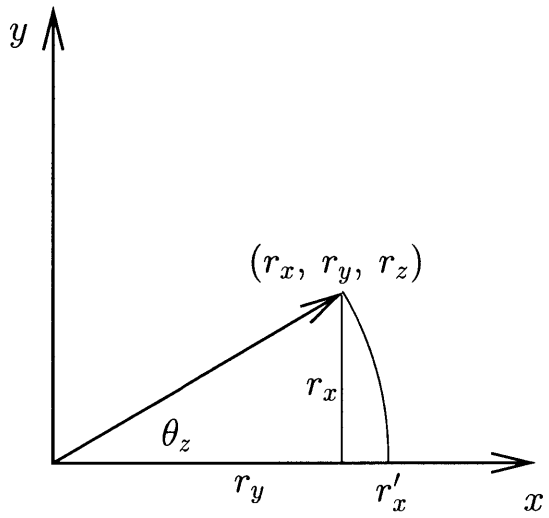


Figure 2-3: θ_z-rotation of finite element

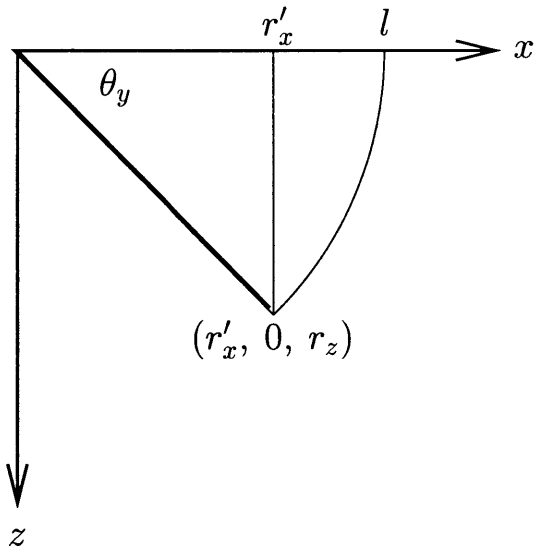


Figure 2-4: θ_y-rotation of finite element

θ_y is more difficult. As seen in Figures 2-3 and 2-4, r'_x is the intermediate x coordinate that the beam must rotate to after its θ_y-rotation. At this point in time,

the beam will lie in the xz plane. Then, θ_z rotates the beam to its final position.

$$r'_x = \frac{r_x}{\cos(\theta_z)} \tag{2.24}$$

$$\theta_y = \text{atan2}(r'_x, r_z) \tag{2.25}$$

All θ's are now fully defined, and can be plugged into the unit rotation equations (2.18) - (2.20) to complete the transformation matrix.

2.3 Force Recovery

Once nodal displacements are known, the internal structure element forces and moments can easily be constructed. By combining Eqns. (2.1) and (2.2), we have an expression for a beam's local forces in terms of its global displacements:

$$\mathbf{F}_\ell = \mathbf{K}_\ell \, \mathbf{T}_{0/1}^{\mathrm{T}} \, \mathbf{x_g} \tag{2.26}$$

First we find displacements in local coordinates:

$$\begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{0/1}^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_{0/1}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{x_{g,1}} \\ \mathbf{x_{g,2}} \end{bmatrix} \tag{2.27}$$

Using a [12x1] matrix to represent beam forces as in Eq. (2.30) would be redundant, since beams must be in static equilibrium. It suffices to only retain the second row:

$$\begin{bmatrix} \mathbf{F_2} \end{bmatrix} = \begin{bmatrix} \mathbf{K_{21}} & \mathbf{K_{22}} \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix}. \tag{2.28}$$

2.4 Element Models

The element stiffness matrix alluded to in Eq. (2.1) contains all the information necessary to describe an arbitrary body, connected to some number of nodes, with a linear elastic stiffness response. Since the script is written for three dimensional space, each node will have six DOF — three translational and three rotational DOF. All beam elements modelled in CoMeT connect two nodes, so $\mathbf{K}_\ell$ must be a [12x12] matrix. For a uniform beam with length l, Young's modulus E, shear modulus G, cross sectional area A, torsional moment of inertia J, and area moments of inertia I_y and I_z for x-z and x-y plane rotations, respectively, the complete element stiffness matrix, if shear deflections are ignored, turns out to be:

$$\begin{bmatrix} F_{x1} \\ F_{y1} \\ F_{z1} \\ M_{x1} \\ M_{y1} \\ M_{z1} \\ F_{x2} \\ F_{y2} \\ F_{z2} \\ M_{x2} \\ M_{y2} \\ M_{z2} \end{bmatrix} = \left[\begin{array}{cccccc|cccccc} \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{l^3} & 0 & 0 & 0 & \frac{6EI_z}{l^2} & 0 & -\frac{12EI_z}{l^3} & 0 & 0 & 0 & \frac{6EI_z}{l^2} \\ 0 & 0 & \frac{12EI_y}{l^3} & 0 & -\frac{6EI_y}{l^2} & 0 & 0 & 0 & -\frac{12EI_y}{l^3} & 0 & -\frac{6EI_y}{l^2} & 0 \\ 0 & 0 & 0 & \frac{GJ}{l} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{l} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{l^2} & 0 & \frac{4EI_y}{l} & 0 & 0 & 0 & \frac{6EI_y}{l^2} & 0 & \frac{2EI_y}{l} & 0 \\ 0 & \frac{6EI_z}{l^2} & 0 & 0 & 0 & \frac{4EI_z}{l} & 0 & -\frac{6EI_y}{l^2} & 0 & 0 & 0 & \frac{2EI_z}{l} \\ \hline -\frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{l^3} & 0 & 0 & 0 & -\frac{6EI_z}{l^2} & 0 & \frac{12EI_z}{l^3} & 0 & 0 & 0 & -\frac{6EI_z}{l^2} \\ 0 & 0 & -\frac{12EI_y}{l^3} & 0 & \frac{6EI_y}{l^2} & 0 & 0 & 0 & \frac{12EI_y}{l^3} & 0 & \frac{6EI_y}{l^2} & 0 \\ 0 & 0 & 0 & -\frac{GJ}{l} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{l} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{l^2} & 0 & \frac{2EI_y}{l} & 0 & 0 & 0 & \frac{6EI_y}{l^2} & 0 & \frac{4EI_y}{l} & 0 \\ 0 & \frac{6EI_z}{l^2} & 0 & 0 & 0 & \frac{2EI_z}{l} & 0 & -\frac{6EI_z}{l^2} & 0 & 0 & 0 & \frac{4EI_z}{l} \end{array}\right] \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \theta_{x1} \\ \theta_{y1} \\ \theta_{z1} \\ x_2 \\ y_2 \\ z_2 \\ \theta_{x2} \\ \theta_{y2} \\ \theta_{z2} \end{bmatrix} \tag{2.29}$$

We can represent Eq. (2.29) in block form as follows:

$$\begin{bmatrix} \mathbf{F_1} \\ \mathbf{F_2} \end{bmatrix} = \begin{bmatrix} \mathbf{K_{11}} & \mathbf{K_{12}} \\ \mathbf{K_{21}} & \mathbf{K_{22}} \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} \tag{2.30}$$

Each column can be thought of as the end node loads required to create unit

displacements in each DOF (corresponding to that column). For example, to stretch node 2 one unit in x (column 7) , node 1 needs to be pulled left with magnitude $\frac{EA}{l}$, while node 2 needs to be pulled right with the same force.

Luckily for the author, the entire [12x12] stiffness matrix need not be constructed analytically for each beam type. Since each beam must be in mechanical equilibrium under any loading condition, the upper half of the matrix depends on the lower half as follows:

$$\mathbf{K_{12}} = -\mathbf{T}_l \, \mathbf{K_{22}} \tag{2.31}$$

$$\mathbf{K_{11}} = -\mathbf{T}_l \, \mathbf{K_{21}} \tag{2.32}$$

Here, $\mathbf{T}_l$ is a 6x6 transformation matrix that expresses the stiffness at node 2 in the frame of node 1. Since both nodes have the same angular orientation, it just represents an x-translation of magnitude l (see section 2.2 for details):

$$\mathbf{T}_l = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -l & 0 & 1 & 0 \\ 0 & l & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.33}$$

We also know from symmetry that

$$\mathbf{K_{21}} = \mathbf{K_{12}^T} \tag{2.34}$$

which can be shown to be equivalent to the statement that solid body motion does not induce internal forces. So the entire stiffness matrix can rapidly be computed once one of the block stiffnesses is known. $\mathbf{K_{22}}$ is the most intuitive block to work with,

since it represents the forces required at node 2 to induce unit motions at node 2, while node 1 can be thought of as grounded.

One more step remains. The "natural" beam deflections one can compute are *compliance* relations — given a unit force, how does the cantilever beam deflect? The responses to each unit force can be assembled into a compliance matrix, and

$$\mathbf{K_{22}} = \mathbf{C_{22}^{-1}} \tag{2.35}$$

Although $\mathbf{K_{22}}$ happens to have a concise analytic form in Eq. (2.29), it does not simplify as well in more general cases, including every case to follow.

In summary, we can compute the entire element stiffness matrix for any beam once deflections under various cantilever loading conditions are determined.

2.4.1 Rectangular Beams

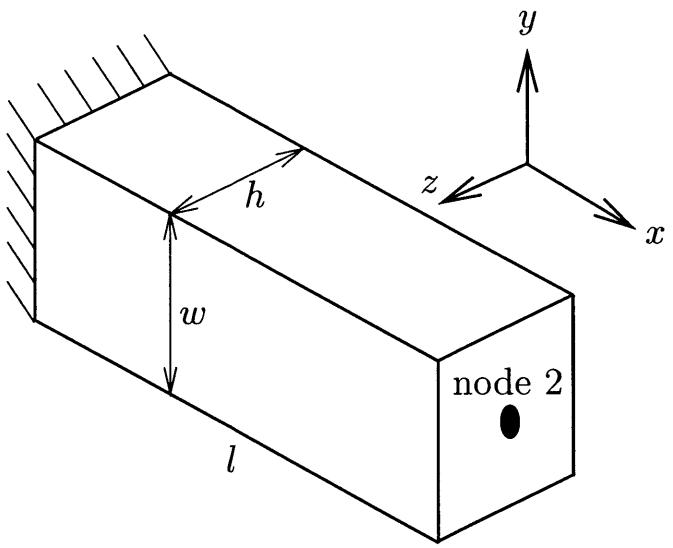


Figure 2-5: Uniform rectangular beam geometry

Rectangular beams, as pictured in Fig. 2-5, are the simplest and most common flexural elements. Since their tensional compliance is typically orders of magnitude smaller than their bending compliance, they provide physically compact constraints

to motion along their length. In addition, the height to width aspect ratio of a beam strongly affects its ratio of in-plane to out-of-plane bending compliance, giving a flexure designer the power to readily achieve desired characteristics. Finally, rectangular beams are usually easy to manufacture by virtue of their simple geometry.

Equation (2.29) already specifies the solution for a straight rectangular beam, with

$$A = h \cdot w \tag{2.36}$$

$$I_y = \tfrac{1}{12} w h^3 \tag{2.37}$$

$$I_z = \tfrac{1}{12} h w^3 \tag{2.38}$$

Courtesy of Young and Budynas [17, p. 401], the torsional moment of inertia is

$$J = ab^3 \left\{ \tfrac{16}{3} - 3.36 \left(\frac{b}{a}\right) \left[1 - \tfrac{1}{12} \left(\frac{b}{a}\right)^4 \right] \right\} \tag{2.39}$$

with a and b the half-height and half-width such that $a > b$.

However, Eq. (2.29) neglects shear deformations, nor does it fit within the framework of the system.

Although seldom emphasized in the engineering curriculum, beams deflect under shear loading, in addition to moment bending. The longer a beam, the less significant the effect, but the critical length that determines if shear can be ignored is not always as short as one might suppose. According to Dowling [11], shear deflection for cross-sectionally symmetric beams may be quantified as follows:

$$\delta = \frac{VL}{GA} \tag{2.40}$$

where V is the shear force and L the length of the beam. The integral equivalent of

Eq. (2.40) is:

$$\delta(x) = \int_0^x \frac{V}{GA}\,\mathrm{d}x' \tag{2.41}$$

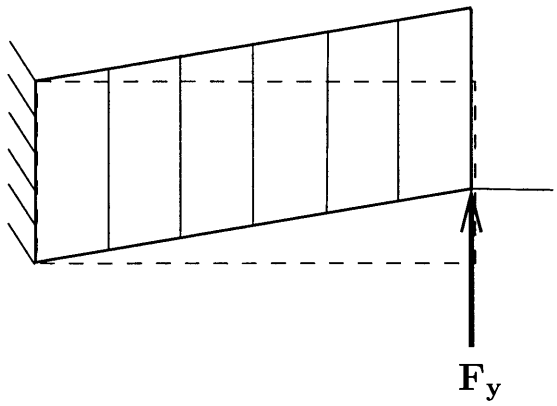


Figure 2-6: Shear deflection mode for a uniform beam

The shear deflection mode is depicted in Figure 2-6. Note that deflection varies linearly with x. Also, although the slope of the upper beam surface is certainly no longer zero, there is no rotation associated with shear. Naturally, shear deflections never occur without moment deflections, since shear is integrated to obtain the moment distribution in a beam.

Since shear deflections are linear, they may be superposed with bending deformations to obtain total deflection (now specializing to loading in the y and z directions):

$$y = \frac{F_y l^3}{3EI_z} + \frac{F_y l}{GA} \tag{2.42}$$

$$z = \frac{F_z l^3}{3EI_y} + \frac{F_z l}{GA} \tag{2.43}$$

Let us take a moment to discuss the significance of shear deflections. The ratio of shear to bending deflection expresses this quantity in dimensionless form.

$$\frac{\delta_{shear}}{\delta_{bending}} = \frac{3EI}{GAL^2} \tag{2.44}$$

Poisson's ratio relates E to G:

$$\frac{E}{G} = 2(1+\nu) \tag{2.45}$$

For a rectangular beam, Eq. (2.44) reduces to

$$\left.\frac{\delta_{shear}}{\delta_{bending}}\right|_{rectangular} = \frac{1+\nu}{2}\left(\frac{h}{L}\right)^2 \tag{2.46}$$

while a round beam (using Eqns (2.50) and (2.52)) becomes

$$\left.\frac{\delta_{shear}}{\delta_{bending}}\right|_{round} = \frac{3(1+\nu)}{4}\left(\frac{h}{L}\right)^2 \tag{2.47}$$

Although correct, Eqns (2.46) and (2.47) do not represent a worst-case scenario. Boundary conditions may exacerbate the effect of shear. For example, the "free but guided" configuration, common in flexures as pictured in Figure 2-7, deflects significantly more when shear is accounted for. This is because slope conditions "stiffen" the bending mode but have no effect on shear.

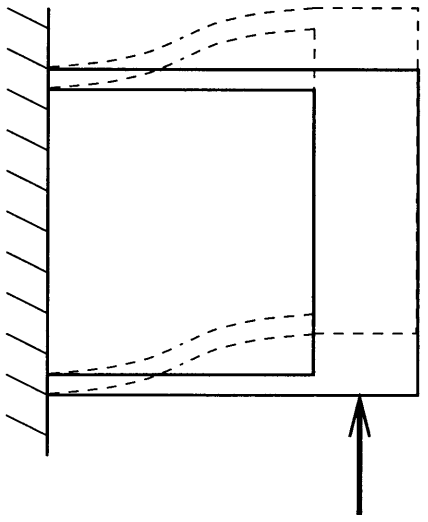


Figure 2-7: Deformation of a parallel beam configuration

If the beams in Figure 2-7 are rectangular, the equivalent of Eq. (2.44) is

$$\left.\frac{\delta_{shear}}{\delta_{bending}}\right|_{free\ but\ guided} = 2\,(1+\nu)\left(\frac{h}{L}\right)^2 \tag{2.48}$$

Shear deflection in Eq. (2.48) is four times greater than in Eq. (2.46) for a given bending deflection.

Figure 2-8: Beam aspect ratio vs. shear to bend deflection ratio

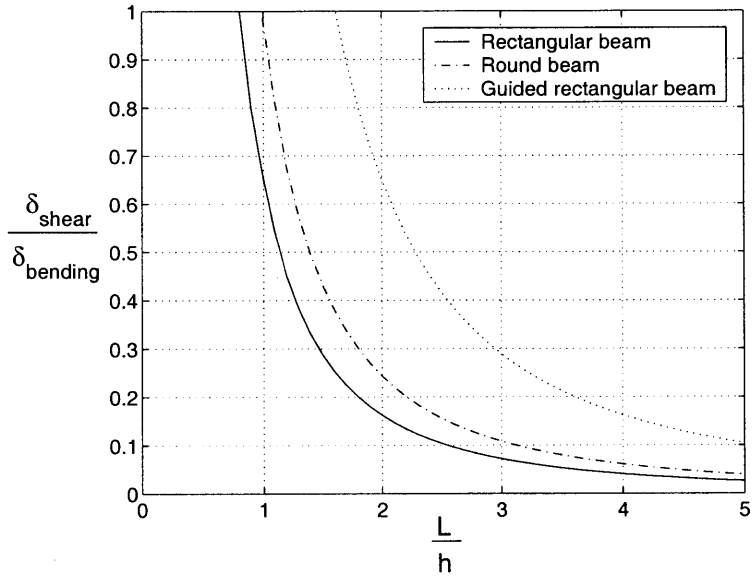


Figure 2-8 graphs Eqns (2.46), (2.47), and (2.48) as a function of the beam aspect ratio for $\nu = 0.3$. Select values of these functions, for the same value of ν, are tabulated in Table 2.1.

Table 2.1: **Beam aspect ratio $\left(\frac{L}{h}\right)$ that results in given shear significance**

$\delta_{shear}/\delta_{bend}$	100%	10%	5%	1%
Rectangular	0.80	2.55	3.61	8.06
Round	0.99	3.12	4.42	9.87
Guided	1.61	5.10	7.21	16.1

Combining Eqns (2.42), (2.43), and the standard cantilever beam equations, we

can construct the compliance matrix:

$$\mathbf{C_{22}} = \begin{bmatrix} \frac{l}{EA} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{l^3}{3EI_z} + \frac{l}{GA} & 0 & 0 & 0 & \frac{l^2}{2EI_z} \\ 0 & 0 & \frac{l^3}{3EI_y} + \frac{l}{GA} & 0 & -\frac{l^2}{2EI_y} & 0 \\ 0 & 0 & 0 & \frac{l}{GJ} & 0 & 0 \\ 0 & 0 & -\frac{l^2}{2EI_y} & 0 & \frac{l}{EI_y} & 0 \\ 0 & \frac{l^2}{2EI_z} & 0 & 0 & 0 & \frac{l}{EI_z} \end{bmatrix} \tag{2.49}$$

The matrix is SPD, and x-y planar motion (x, y, θ_z) is de-coupled from out-of-plane forcing and vice-versa.

2.4.2 Round Beams

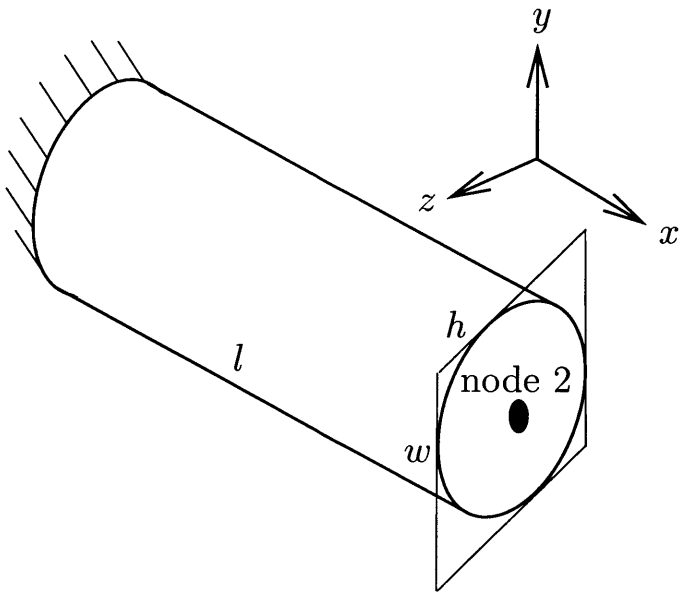


Figure 2-9: Uniform round beam geometry

Round beams, pictured in Fig 2-9, are also fairly common structural elements. They are often easy to manufacture, depending on the field of application. Circular beams have the property that each outer fiber is stressed by the same amount

when torsional loads are applied, allowing more energy storage before yielding than rectangular beams do for a given mass and length. Beams of elliptical cross section may prove useful where loading is predominantly torsional, but a significant bending moment must also be supported.

Analytically, round beams are very similar to rectangular beams, but area properties differ. We can use Eq. (2.49) with the following definitions:

$$A = \frac{\pi}{4} hw \tag{2.50}$$

$$I_y = \frac{\pi}{64} wh^3 \tag{2.51}$$

$$I_z = \frac{\pi}{64} hw^3 \tag{2.52}$$

Again citing Young and Budynas [17, p. 401], the torsional moment of inertia is

$$J = \frac{\pi w^3 h^3}{16\left(w^2 + h^2\right)} \tag{2.53}$$

2.4.3 Tapered Beams

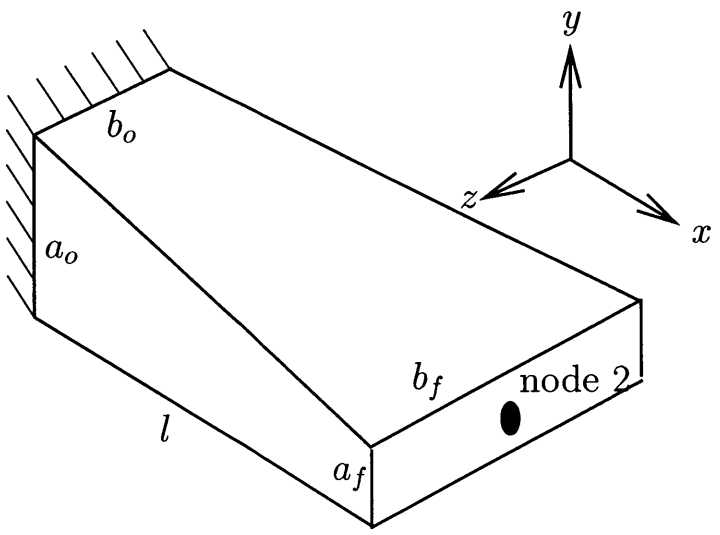


Figure 2-10: Doubly tapered beam geometry

For tapered beams, as depicted in Figure 2-10, height and width vary with x. Since tapered beam geometry involves more parameters, it gives flexure designers more inputs to optimize. For instance, if loading conditions are known and fairly unidirectional, the beam taper can be chosen to maximize energy stored for a given stiffness, mass, and length. The maximum tensile stress in a given beam cross section is

$$\sigma_{max} = \frac{Mc}{I} \tag{2.54}$$

if I is varied inversely with M, stress distributions may be equalized. The solver module described in section 3.1 can be used to help a designer optimize geometry under three-dimensional loading.

Although the compliance of tapered beams is more analytically involved, it is conceptually no different from the beams discussed previously.

Tension

For a beam with linear stress-strain behavior, tensional extension can be expressed in integral form:

$$x(l) = \int_0^l \frac{F_x}{EA(x)}\,\mathrm{d}x \tag{2.55}$$

Eq. (2.55) loses applicability with strong degrees of taper, since the implicitly assumed stress distribution would leave the surface in disequilibrium. Subsection 5.1.4 discusses Eq. (2.55)'s range of validity.

For convenience, we define attenuation factors α and β:

$$\alpha \equiv \frac{a_f - a_o}{a_o\, l}; \qquad \beta \equiv \frac{b_f - b_o}{b_o\, l} \tag{2.56}$$

Using these terms, Eq. (2.55) may be reexpressed as

$$x(l) = \frac{F}{EA_o}\int_0^l \frac{1}{(1+\alpha x)(1+\beta x)}\,\mathrm{d}x \tag{2.57}$$

where A_o is the cross-sectional area at $x = 0$. The integral evaluates to

$$\frac{\log(1+\alpha l) - \log(1+\beta l)}{\alpha - \beta} \tag{2.58}$$

For the sake of constructing the compliance matrix, we need a coefficient C_{F_x} such that

$$x = C_{F_x} F_x \tag{2.59}$$

Combining Eqns (2.57) and (2.58), it is apparent that

$$C_{F_x} = \frac{1}{EA_o}\left[\frac{\log(1+\alpha l) - \log(1+\beta l)}{\alpha - \beta}\right] \tag{2.60}$$

Despite the ostensive division by zero in Eq. (2.60) as α approaches β, Eq. (2.60) is well behaved, as one might expect on physical grounds. If $\alpha = \beta$, it reaches the following singular limit:

$$C_{F_x} = \frac{1}{EA_o}\left(\frac{1}{1+\alpha l}\right) \tag{2.61}$$

Shear

Equation (2.41) also holds for tapered beams. Evaluated at $x = l$, it becomes

$$\delta = \int_0^l \frac{F}{GA}\,\mathrm{d}x \tag{2.62}$$

Since Eq. (2.62) has the same structure as Eq. (2.55), the shear contribution to deflection can be stated quite simply:

$$C_{F_y\ shear} = C_{F_z\ shear} = C_{F_x} \cdot \frac{G}{E} \tag{2.63}$$

From now on, we can consider our task to be finding the coefficients of $\mathbf{C_{22}}$ as implicitly defined in Eq. (2.64) below. Note that C_{F_x} and C_{F_y} need only account

for bending deflections; the shear components, as defined by Eq. (2.63), are already included. Also, ζ is used to denote angular deflection coefficients.

$$\mathbf{C_{22}} = \begin{bmatrix} C_{F_x} & 0 & 0 & 0 & 0 & 0 \\ 0 & C_{F_y} + \frac{G}{E}C_{F_x} & 0 & 0 & 0 & C_{M_z} \\ 0 & 0 & C_{F_z} + \frac{G}{E}C_{F_x} & 0 & -C_{M_y} & 0 \\ 0 & 0 & 0 & \zeta_{M_x} & 0 & 0 \\ 0 & 0 & -\zeta_{F_z} & 0 & \zeta_{M_y} & 0 \\ 0 & \zeta_{F_y} & 0 & 0 & 0 & \zeta_{M_z} \end{bmatrix} \tag{2.64}$$

Cantilever moment loading

Since our cantilever is symmetrically tapered, the moment-curvature relation holds:

$$M = \frac{EI}{\rho} \tag{2.65}$$

For small deflections, Eq. (2.65) becomes

$$M_y = EI_y y'' \tag{2.66}$$

$$M_z = EI_z z'' \tag{2.67}$$

Since y and z are analogous, let us just consider the planar case for now. By integrating, substituting Eqns (2.38) and (2.56), and moving invariant quantities outside the integral, Eq. (2.66) becomes

$$y' = \frac{M_z}{E\frac{1}{12}b_o a_o^3} \int_o^x \frac{1}{(1+\beta x)(1+\alpha x)^3}\, \mathrm{d}x \tag{2.68}$$

Using the definition

$$I_{z_o} \equiv \tfrac{1}{12}\, b_o a_o^3\,, \tag{2.69}$$

upon evaluation of the integral, we find

$$\zeta_{M_z} = \frac{1}{EI_{z_o}} \left\{ \frac{\beta^2 \left[\ln(1+\beta l) - \ln(1+\alpha l)\right]}{(\beta-\alpha)^3} + \frac{\beta \left[(1+\alpha\, l)^{-1} - 1\right]}{(\beta-\alpha)^2} + \frac{(1+\alpha\, l)^{-2} - 1}{2\,(\beta-\alpha)} \right\} \tag{2.70}$$

$$\text{If } \alpha = \beta \;\; : \;\; \zeta_{M_z} = \frac{1}{3EI_{z_o}} \left\{ -\frac{1}{(1+\alpha\, l)^3\, \alpha} + \frac{1}{\alpha} \right\} \tag{2.71}$$

$$\text{If } \alpha = \beta = 0 \;\; : \;\; \zeta_{M_z} = \frac{l}{EI_{z_o}} \tag{2.72}$$

To get y-deflection due to moment loading, we integrate once more:

$$y = \int_o^l y' \, \mathrm{d}x \tag{2.73}$$

The result is

$$C_{M_z} = \frac{1}{EI_{z_o}} \left\{ \frac{\beta\,(1+\beta\, l)\left[\ln(1+\beta l) - \ln(1+\alpha l)\right]}{(\beta-\alpha)^3} - \frac{\beta\, l}{(\beta-\alpha)^2} - \frac{\alpha\, l^2}{2\,(\beta-\alpha)\,(1+\alpha\, l)} \right\}; \tag{2.74}$$

$$\text{If } \alpha = \beta \;\; : \;\; C_{M_z} = \frac{1}{6EI_{z_o}} \left\{ \frac{l^2\,(3+2\,\alpha\, l)}{(1+\alpha\, l)^2} \right\} . \tag{2.75}$$

ζ_{M_y} and C_{M_y} are analogous to ζ_{M_z} and C_{M_z}, with α, β interchanged, and I_{y_o} taking the place of I_{z_o}.

Cantilever force loading

Force loading is, once again, solved by integrating moment-curvature. This time, the moment in Eq. (2.65) is no longer constant; it varies linearly with x and vanishes at the tip.

$$M = F\,(x - l) \tag{2.76}$$

We could obtain ζ_{F_y} by integrating

$$y' = \frac{F_y}{EI_{z_o}} \int_0^l \frac{l \, \mathrm{d}x}{(1+\beta x)(1+\alpha x)^3} - \frac{F_y}{EI_{z_o}} \int_0^l \frac{x \, \mathrm{d}x}{(1+\beta x)(1+\alpha x)^3}, \tag{2.77}$$

but we already know that ζ_{F_y} must equal C_{M_z} for $\mathbf{C_{22}}$ to remain symmetric. Equation (2.77) may be used to check the correctness of Eq. (2.74).

To obtain C_{F_y}, we once again apply Eq. (2.73).

$$C_{F_y} = C_{M_z} l - \frac{1}{EI_{z_o}} \left\{ \frac{(1+\beta l)\left[\ln(1+\beta l) - \ln(1+\alpha l)\right]}{(\alpha-\beta)^3} + \frac{l}{(\alpha-\beta)^2} - \frac{l^2}{2\,(\alpha-\beta)\,(1+\alpha l)} \right\} \tag{2.78}$$

$$\text{If } \alpha = \beta \;:\; C_{F_y} = \frac{1}{3EI_{z_o}} \left\{ \frac{l^3}{(1+\alpha\, l)} \right\} \tag{2.79}$$

As seen before for ζ_{M_y} and C_{M_y}, ζ_{F_z} and C_{F_z} are analogous to ζ_{F_y} and C_{F_y}.

Torsion

For torsional compliance of tapered rectangular beams,

$$\frac{\theta_x}{M_x} = \zeta_{M_x} = \int_0^l \frac{1}{GJ} \, \mathrm{d}x \tag{2.80}$$

with J given by Eq. (2.39). Since J is in the denominator and piecewise continuous, a closed form solution is difficult to attain. However, Eq. (2.80) may simply be integrated numerically once beam geometry is specified. If future work demanded an analytic equation, one approach may include scaling Eq. (2.53) to approximately match the inertia of a rectangular beam, then integrating the much simpler expression.

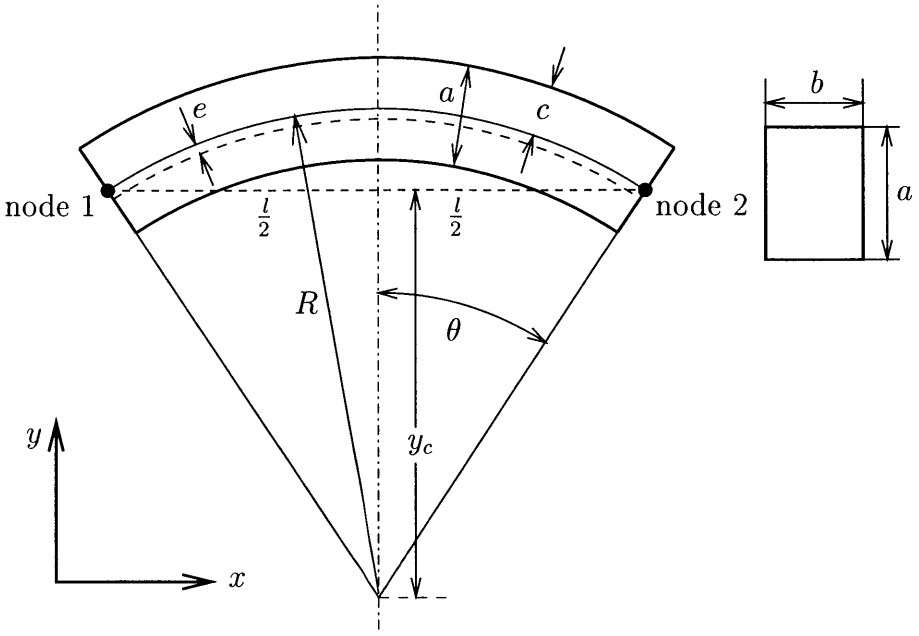


Figure 2-11: Curved beam geometry for planar loads

2.4.4 Curved Beams

Curved beams, as pictured in Fig 2-11, are omnidirectionally compliant to first order, assuming their included angle 2θ sufficiently distinguishes them from rectangular beams. They may provide smooth transitions between beams aligned to different directions, reducing the potential for stress concentrations. By nature, they fit compactly around cylindrical geometry, which is inherent to many flexural systems. The beam aspect ratio may be varied, just as with rectangular beams, to control the relative significance of planar and non-planar stiffnesses. Slightly curved beams may be used in designs where the direction of buckling need be controlled. Note that y_c must be positive and beams must arch clockwise in their local frame. This does not pose a limitation to the user, however, since he or she need only interchange the order of a beam's end nodes to have it appear counterclockwise in the global frame.

Curved beam compliance matrices are less sparse than Eq. (2.64), since each type of loading tends to perturb three degrees of freedom. We therefore generalize

Eq. (2.64) and write

$$\mathbf{C_{22}} = \begin{bmatrix} C_{\mathrm{x}F_x} & C_{\mathrm{x}F_y} & 0 & 0 & 0 & C_{\mathrm{x}M_z} \\ C_{\mathrm{y}F_x} & C_{\mathrm{y}F_y} & 0 & 0 & 0 & C_{\mathrm{y}M_z} \\ 0 & 0 & C_{F_z} & C_{M_x} & C_{M_y} & 0 \\ 0 & 0 & \zeta_{\mathrm{x}F_z} & \zeta_{\mathrm{x}M_x} & \zeta_{\mathrm{x}M_y} & 0 \\ 0 & 0 & \zeta_{\mathrm{y}F_z} & \zeta_{\mathrm{y}M_x} & \zeta_{\mathrm{y}M_y} & 0 \\ \zeta_{F_x} & \zeta_{F_y} & 0 & 0 & 0 & \zeta_{M_z} \end{bmatrix} \tag{2.81}$$

In-plane geometry is pictured in Figure 2-11. Note that

$$y_c = \sqrt{R^2 - \left(\tfrac{1}{2}\, l\right)^2} \tag{2.82}$$

$$\theta = \sin^{-1}\left(\frac{l}{2R}\right) \tag{2.83}$$

Young and Budynas [17] have analyzed curved beam geometry at great length. Their equations are adopted and modified where nomenclature or format differ from those of this thesis. Most notably, the Young and Budynas equations essentially give the coefficients of $\mathbf{C_{11}}$. Due to left-right symmetry, these differ only in sign from those of $\mathbf{C_{22}}$. But first, a few auxiliary variables demand definition. The ratio $\frac{e}{R}$ quantifies the shift of the neutral axis toward the concave side of the beam, as pictured.

$$e/R = 1 - \frac{a}{R \ln\left(\frac{2R/a+1}{2R/a-1}\right)} \tag{2.84}$$

Poisson's ratio ν can easily be isolated from Eq. (2.45), and we may define strain constants K_1 and K_2.

$$\nu = \frac{E}{2G} - 1 \tag{2.85}$$

$$K_2 = 1 - e/R \tag{2.86}$$

$$K_1 = K_2 + \tfrac{12}{5}\,(1+\nu)\, e/R \tag{2.87}$$

The second area moment of inertia is defined as before,

$$I_z = \tfrac{1}{12} b a^3 \tag{2.88}$$

For convenience, we abbreviate trigonometric functions of θ:

$$s \equiv \sin\theta \qquad\qquad c \equiv \cos\theta \tag{2.89}$$

Now the planar coefficients of the compliance matrix can be defined.

$$C_{\mathrm{x}F_x} = \frac{R^3}{EI_z}\left[2\theta c^2 + K_1(\theta - sc) - K_2(2sc)\right] \tag{2.90}$$

$$C_{\mathrm{x}F_y} = C_{\mathrm{y}F_x} = \frac{R^3}{EI_z}\left[-2\theta sc + K_2(2s^2)\right] \tag{2.91}$$

$$C_{\mathrm{x}M_z} = \zeta_{F_x} = \frac{R^2}{EI_z}\left[-2\theta c + K_2(2s)\right] \tag{2.92}$$

$$C_{\mathrm{y}F_y} = \frac{R^3}{EI_z}\left[2\theta s^2 + K_1(\theta + sc) - 2K_2 sc\right] \tag{2.93}$$

$$C_{\mathrm{y}M_z} = \zeta_{F_y} = \frac{R^2}{EI_z}\left[2\theta s\right] \tag{2.94}$$

$$\zeta_{M_z} = \frac{R}{EI_z}\left[-2\theta\right] \tag{2.95}$$

Young and Budynas [17] employ different notation for out-of-plane analysis. Adjustments have been made to maintain consistency.

We now define β, a nondimensional parameter that quantifies a beam's tendency to twist, relative to its ability to bend.

$$\beta = \frac{EI_y}{GJ} \tag{2.96}$$

with J given in Eq. (2.39).

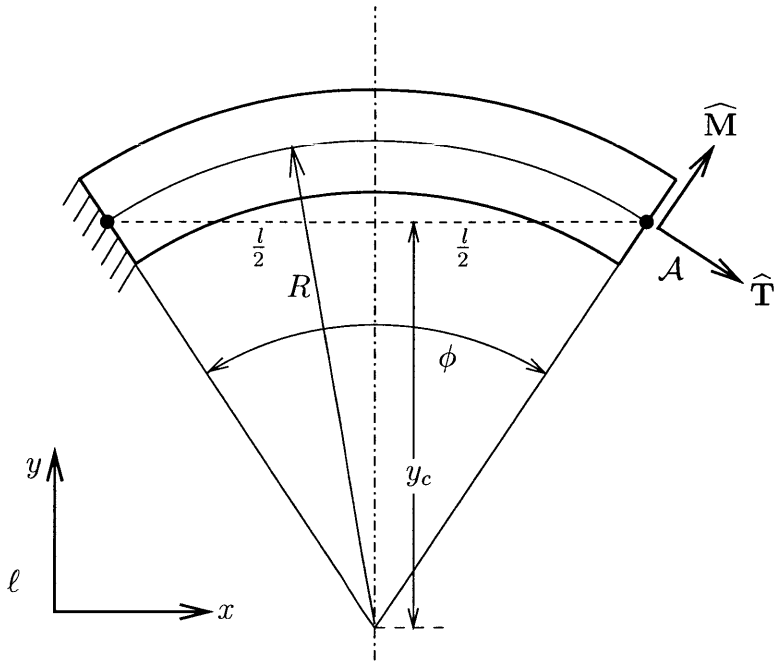


Figure 2-12: Curved beam geometry for non-planar loads

In Figure 2-12 we see that ϕ is the whole subtended angle:

$$\phi = 2\theta \tag{2.97}$$

Constants C_i below are used in Young and Budynas to simplify expressions. Although all C_i could easily be defined explicitly, back-referencing to C_2 and C_5 reduces the number of floating point operations (flops) required during execution of the MATLAB script.

$$C_2 = \frac{1+\beta}{2}(\phi\cos\phi - \sin\phi) \tag{2.98}$$

$$C_5 = -\frac{1+\beta}{2}\phi\sin\phi \tag{2.99}$$

$$\tag{2.100}$$

$$C_1 = -C_5 - \beta(1-\cos\phi) \tag{2.101}$$

$$C_3 = \beta(\sin\phi - \phi) - C_2 \tag{2.102}$$

$$C_4 = C_2 + \sin\phi \tag{2.103}$$

$$C_8 = -C_2 - \beta \sin\phi \tag{2.104}$$

$$\tag{2.105}$$

Deflections are expressed in a coordinate frame $\mathcal{A}$ tangent to the undeformed beam, as depicted in Fig. 2-12.

$$C_{ZW} = \frac{R^3}{EI_y}\left[C_1 \sin\phi - C_2(1-\cos\phi) - C_3\right] \tag{2.106}$$

$$C_{ZM} = \zeta_{\Theta W} = -\frac{R^2}{EI_y}\left[C_1 \cos\phi - C_2 \sin\phi\right] \tag{2.107}$$

$$C_{ZT} = \zeta_{\Psi W} = \frac{R^2}{EI_y}\left[C_1 \sin\phi + C_2 \cos\phi\right] \tag{2.108}$$

$$C_{ZM} = -\frac{R^2}{EI_y}\left[C_4 \sin\phi - C_5(1-\cos\phi) - C_1\right] \tag{2.109}$$

$$\zeta_{\Theta M} = -\frac{R}{EI_y}\left[C_5 \sin\phi - C_4 \cos\phi\right] \tag{2.110}$$

$$\zeta_{\Theta T} = \zeta_{\Psi M} = -\frac{R}{EI_y}\left[C_4 \sin\phi + C_5 \cos\phi\right] \tag{2.111}$$

$$C_{ZT} = -\frac{R^2}{EI_y}\left[C_5 \sin\phi - C_8(1-\cos\phi) - C_2\right] \tag{2.112}$$

$$\zeta_{\Theta T} = \frac{R}{EI_y}\left[C_5 \cos\phi - C_8 \sin\phi\right] \tag{2.113}$$

$$\zeta_{\Psi T} = -\frac{R}{EI_y}\left[C_8 \cos\phi + C_5 \sin\phi\right] \tag{2.114}$$

$$\tag{2.115}$$

$$\zeta_{\mathcal{A}} = \begin{bmatrix} C_{ZW} & C_{ZM} & C_{ZT} \\ \zeta_{\Theta W} & \zeta_{\Theta M} & \zeta_{\Theta T} \\ \zeta_{\Psi W} & \zeta_{\Psi M} & \zeta_{\Psi T} \end{bmatrix} \tag{2.116}$$

As evident in Figure (2-12), conversion to the local (ℓ) frame requires coordinate rotations:

$$\zeta_\ell = \mathbf{R}_{\ell/\mathcal{A}}\, \zeta_{\mathcal{A}}\, \mathbf{R}_{\ell/\mathcal{A}}^{\mathbf{T}} \tag{2.117}$$

$$\mathbf{R}_{\ell/\mathcal{A}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & s \\ 0 & -s & c \end{bmatrix} \tag{2.118}$$

$\mathbf{R}_{\ell/\mathcal{A}}$ happens to resemble $\mathbf{R}_{\mathbf{x}}^{\mathbf{T}}$ from Eq. (2.18) because it performs a similar operation; however, the two are not formally identical.

The elements of ζ_ℓ complete the non-planar region of $\mathbf{C_{22}}$.

$$\zeta_\ell = \begin{bmatrix} C_{F_z} & C_{M_x} & C_{M_y} \\ \zeta_{\mathrm{x}F_z} & \zeta_{\mathrm{x}M_x} & \zeta_{\mathrm{x}M_y} \\ \zeta_{\mathrm{y}F_z} & \zeta_{\mathrm{y}M_x} & \zeta_{\mathrm{y}M_y} \end{bmatrix} \tag{2.119}$$

Incidentally, $\zeta_{\mathrm{x}M_y}$ and $\zeta_{\mathrm{y}M_x}$ are identically zero. Coefficients $\zeta_{\Theta T}$ and $\zeta_{\Psi M}$ need still be found, however, so that the remaining coefficients of ζ_ℓ may be computed.

The equations listed are now sufficient to find curved beam deformations under arbitrary loading conditions. FEA results are presented for comparison and validation in subsection 5.1.4.

2.4.5 Rigid Plates

Many structures are only designed to flex in a relatively small material volume, with large pieces remaining effectively rigid. This poses a problem inherent to finite element analysis, since stiffness matrices tend to become ill-conditioned when large stiffness variations exist. In addition, default meshing routines usually create elements of similar size throughout a structure, wasting computing power on regions that have little influence on flexural kinematics. Advanced users may be able to customize the grid layout to reflect regions of significance in a structure, but doing so takes time and often requires changes to the CAD model. CoMeT circumvents both conditioning and element count problems by allowing the user to define infinitely rigid connections between elements. Although these connections are purely mathematical constraints between nodes, to the user they appear just like rigid plate elements. Unlike other

elements, however, plates reduce the size of $\mathbf{K_G}$ by eliminating the superfluous degrees of freedom of the constrained nodes. In other words, near-singular constraints are approximated as precisely singular, but the redundant constraints are removed prior to inversion.

The basic procedure outlining plate implementation is as follows:

1. $\mathbf{K_G}$ is reduced to $\mathbf{K'_G}$ to allow fewer DOF

2. $\mathbf{K'_G}$ is inverted to yield $\mathbf{C'_G}$

3. $\mathbf{C'_G}$ is expanded to $\mathbf{C_G}$, which has the original size of $\mathbf{K_G}$

During the reduction phase, all stiffnesses relating to a given plate are reflected to the frame of the first node in that plate. Then, the rows and columns of the other nodes are deleted.

Let $\mathbf{T_{i/j}}$ be a transformation matrix relating the global frame at node j to that of node i, both contained in a plate. In the spirit of Eqns (2.2)-(2.5),

$$\mathbf{x_j} = \mathbf{T_{i/j}^T}\,\mathbf{x_i} \tag{2.120}$$

$$\mathbf{x_i} = \left(\mathbf{T_{i/j}^T}\right)^{-1}\mathbf{x_j} \tag{2.121}$$

$$\mathbf{F_i} = \mathbf{T_{i/j}}\,\mathbf{F_j} \tag{2.122}$$

$$\mathbf{F_j} = \mathbf{T_{i/j}^{-1}}\,\mathbf{F_i} \tag{2.123}$$

$\mathbf{T_{i/j}}$ is a [6x6] containing pure translations and no rotations. Mathematically, $\mathbf{T_{i/j}}$ is defined just like $\mathbf{T_{0/1}}$ in Eq. (2.15), with translation component $\mathbf{C}(\mathbf{r_i})$ given by Eq. (2.124):

$$\mathbf{C}(\mathbf{r_i}) = \begin{bmatrix} 0 & -(r_{z,j}-r_{z,i}) & r_{y,j}-r_{y,i} \\ r_{z,j}-r_{z,i} & 0 & -(r_{x,j}-r_{x,i}) \\ -(r_{y,j}-r_{y,i}) & r_{x,j}-r_{x,i} & 0 \end{bmatrix} \tag{2.124}$$

Next, we make use of $\mathbf{K_{ij}}$, whose meaning becomes clear in the following equation:

$$\mathbf{F_i} = \mathbf{K_{ij}}\,\mathbf{x_j} \tag{2.125}$$

Substituting 2 for i and 1 for j in Eq. (2.125), we have an expression for the stiffness components in the row of node 2:

$$\mathbf{F_2} = \mathbf{K_{2i}}\,\mathbf{x_i} \tag{2.126}$$

Likewise, with $i = 1$, $j = 2$, Eq. (2.123) becomes

$$\mathbf{F_2} = \mathbf{T}_{1/2}^{-1}\,\mathbf{F_1} \tag{2.127}$$

We can combine Eqns (2.126) and (2.127) to generate an equation that resembles the definition of $\mathbf{K_{1i}}$:

$$\mathbf{F_1} = \left(\mathbf{T}_{1/2}\,\mathbf{K_{2i}}\right)\mathbf{x_i} \tag{2.128}$$

The term in parentheses is the stiffness response of node 2 projected to frame 1. If we add it to the original stiffness for each i, nodes 1 and 2 are now represented by a single set of forces.

$$\mathbf{K'_{1i}} = \mathbf{K_{1i}} + \mathbf{T}_{1/2}\,\mathbf{K_{2i}} \tag{2.129}$$

The operation is best understood visually. Figure 2-13 illustrates the implementation of Eq. (2.129).

By similar logic, Eqns (2.125) and (2.120) can be used to constrain $\mathbf{x_2}$ to $\mathbf{x_1}$:

$$\mathbf{K'_{i1}} = \mathbf{K_{i1}} + \mathbf{K_{i2}}\mathbf{T}_{\mathbf{i2}}^{\mathrm{T}} \tag{2.130}$$

Note that in Eq. (2.130), in the row corresponding to node 1, $\mathbf{K_{i2}}$ would actually be $\mathbf{K'_{i2}}$, since that row has already been modified. When the redundant rows and

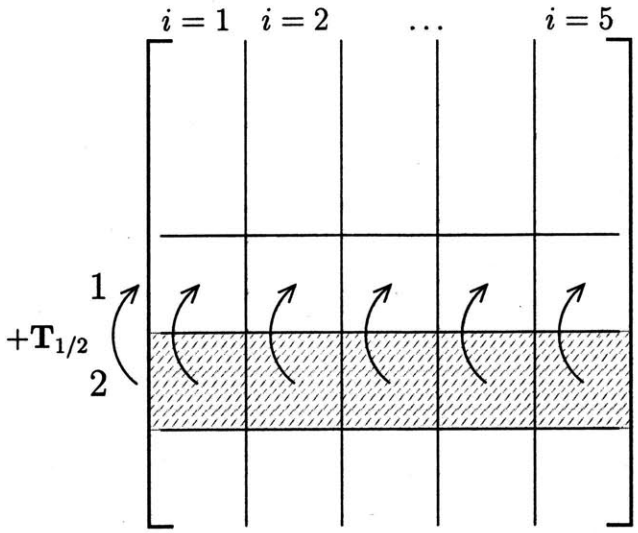


Figure 2-13: Stiffness matrix row reduction

columns are eliminated, $\mathbf{K_G}$ behaves as if nodes 1 and 2 are rigidly connected, while retaining symmetric positive definite properties. The column reduction operation is pictured in Figure 2-14.

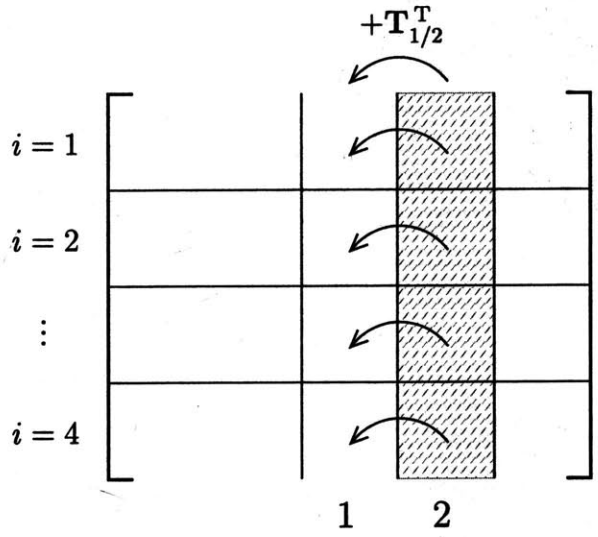


Figure 2-14: Stiffness matrix column reduction

It also warrants mentioning that any stiffness contributions from beams connected at both ends to the same plate have no effect on $\mathbf{K'_G}$. A formal proof is cumbersome,

but the cancellation is essentially a consequence of the minus signs in Eqns (2.31) and (2.32).

After creation via $\mathbf{K'_G}$ inversion, $\mathbf{C'_G}$ can be re-expanded to allow forcing at node 2 and displacement observations at node 1. As one might expect, $\mathbf{C_{ij}}$ relates $\mathbf{x_i}$ to $\mathbf{F_j}$:

$$\mathbf{x_i} = \mathbf{C_{ij}}\,\mathbf{F_j} \tag{2.131}$$

Using a modified version of Eqns (2.131) and (2.121),

$$\mathbf{C_{2i}} = \mathbf{T}_{1/2}^{\mathrm{T}}\mathbf{C_{1i}} \tag{2.132}$$

Columns can be expanded similarly:

$$\mathbf{C_{i2}} = \mathbf{C_{i1}}\mathbf{T}_{1/2} \tag{2.133}$$

The new $\mathbf{C_G}$ behaves just like the original $\mathbf{C_G}$ would in the limit that nodes in a plate are connected by infinitely rigid beams. It is singular, since combinations of internal plate forces exist that create no nodal displacements, or, more obviously, since the rows of node 2 are linear combinations of the rows of node 1 (the same goes for the columns).

A few words on the implementation of plates in CoMeT. Before $\mathbf{K_G}$ is ever constructed, plates are checked for overlap. If any nodes are concurrently on more than one plate, the plates are merged. Similarly, if nodes are simultaneously grounded and part of a plate, the rest of the plate is merged with ground. Then, nodes are renumbered so that the order of nodes is

1. Standard independent nodes
2. Nodes on plate 1, plate 2, ...
3. Grounded nodes

This greatly facilitates both $\mathbf{K_G}$ reduction and $\mathbf{C_G}$ expansion. Figure 2-15 shows the algorithm CoMeT uses to extract $\mathbf{C_G}$ from $\mathbf{C'_G}$. Fields of light gray shading with arrows pointing to them are computed using the appropriate row or column expansion formula, while other fields are copied directly from $\mathbf{C'_G}$.

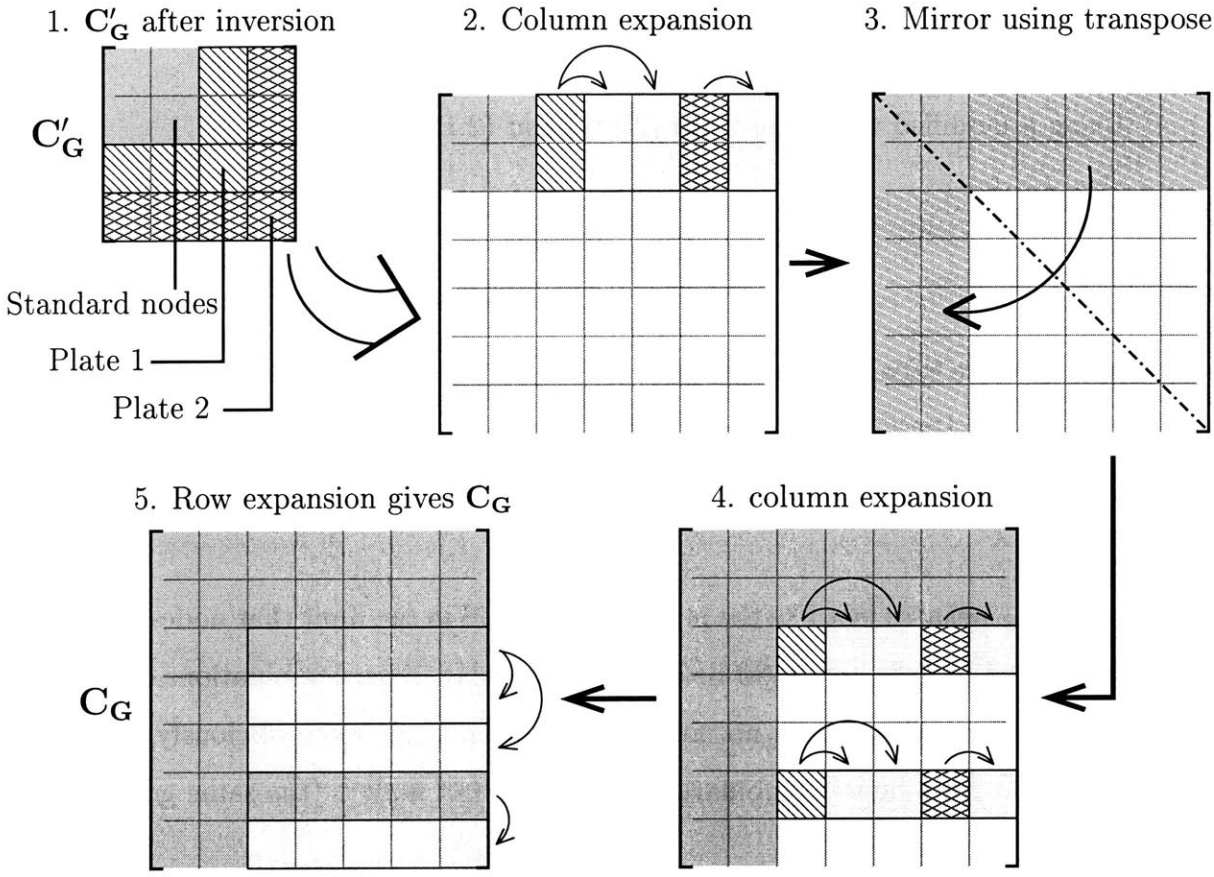


Figure 2-15: $\mathbf{C'_G}$ expansion algorithm

2.5 Maximum Element Stresses

CoMeT uses the von Mises stress criterion to convert general stress states into an

equivalent tensile stress that may be used to predict material failure.

$$\sigma_{\text{von Mises}} = \sqrt{\frac{1}{2}\left[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2\right] + \frac{3}{2}\left(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2\right)} \quad (2.134)$$

As indicated by the presence of shear stresses, Eq. (2.134) does not require that principal stress be found and used; $\sigma_{\text{von Mises}}$ is invariant, and constituent stresses may be expressed in any coordinate frame.

In beams, tensile stress and circumferential shear stress are the dominant components, and Eq. (2.134) reduces to

$$\sigma_{\text{von Mises, beams}} = \sqrt{\sigma_x^2 + \frac{3}{2}\tau_{rt}^2} \quad (2.135)$$

The location of the highest stress in the material, and the subsequent expressions for σ_x and τ_{rt}, vary with beam type and load state.

2.5.1 Rectangular Beams

In rectangular beams, bending stress increases linearly with distance from the neutral axis[1], and tensile stress is uniform, so the highest tensile stress must occur on one of the edges of the beam. There is also a parabolically varying stress distribution due to shear forces, but it is much smaller than tensile stresses, and takes a maximum at the beam center, where tensile stresses are absent. Shear force stresses are therefore neglected in CoMeT's analysis.

Torsion-induced circumferential shear stress takes a maximum at the *innermost* point on the beam surface, which only coincides with the location of the maximum tensile stress in special cases. CoMeT conservatively assumes that maximum tension and shear occur at the same point, an approximation that only leads to appreciable error in rare states of three dimensional loading.

[1]The neutral axis should really be called the neutral plane, but "axis" is the common engineering term.

The moment distribution along the length of a beam under end loading must be linear, so the maximum stress must be in one of the corners. Under combined planar bending, non-planar bending, and tension, at least one of the corners will constructively add all three effects, and since the yield stress magnitude is presumably identical in tension and compression, the absolute value of each load is applied to obtain the highest stress at a cross section.

$$\sigma_{x,\,\max} = \frac{|M_z|\,w}{2\,I_z} + \frac{|M_y|\,h}{2\,I_y} + \frac{|F_x|}{A} \tag{2.136}$$

Torsional shear is given by Young and Budynas [17, p. 401]:

$$\tau_{rt,\,\max} = \frac{3M_x}{8ab^2}\left[1 + 0.6095\,\frac{b}{a} + 0.8865\left(\frac{b}{a}\right)^2 - 1.8023\left(\frac{b}{a}\right)^3 + 0.9100\left(\frac{b}{a}\right)^4\right] \tag{2.137}$$

where a and b the same as in Eq. (2.39). Eqns (2.136) and (2.137) are evaluated and combined via Eq. (2.135) at each end of the beam, and the larger of the two values noted as the maximum von Mises stress.

2.5.2 Round Beams

Round beams are similar to rectangular beams, but the location of maximum stress is now somewhere along the perimeter of the beam end, where

$$\sigma_x = \frac{|M_z|\,y}{I_z} + \frac{|M_y|\,z}{I_y} + \frac{|F_x|}{A}. \tag{2.138}$$

Since the beam is elliptical, y and z are constrained as follows:

$$\frac{y^2}{\left(\frac{w}{2}\right)^2} + \frac{z^2}{\left(\frac{h}{2}\right)^2} = 1 \tag{2.139}$$

Substituting

$$\alpha = \frac{|M_z|}{I_z}, \qquad \beta = \frac{|M_y|}{I_y} \tag{2.140}$$

and maximizing Eq. (2.138) reveals the critical location:

$$y = \frac{\alpha w^2}{2\sqrt{\alpha^2 w^2 + \beta^2 h^2}} \tag{2.141}$$

$$z = \frac{h}{2}\sqrt{1 - \left(\frac{2y}{w}\right)^2} \tag{2.142}$$

Young and Budynas[17, p. 401] give the expression for torsional shear in a round beam:

$$\tau_{rt,\,\max} = \frac{2M_x}{\pi a b^2} \tag{2.143}$$

The equations in this section are evaluated in the proper sequence and the larger of the two end stresses chosen to predict the point of initial yielding.

2.5.3 Curved Beams

In curved beams, the neutral axis shifts toward the inside of the beam, and stress varies nonlinearly in a cross section. Young and Budynas [17, p. 304] give the resulting planar stress concentration factor at the inner edge of the beam, k_c:

$$R/c = \frac{2R}{w} \tag{2.144}$$

$$e/c = R/c - \frac{2}{\ln\left(\frac{R/c+1}{R/c-1}\right)} \tag{2.145}$$

$$k_c = \frac{1}{3e/c}\,\frac{1 - e/c}{R/c - 1} \tag{2.146}$$

Eq. (2.136) can thus be modified to approximate the maximum stress in a curved beam cross section:

$$\sigma_{x,\,\max} = k_c \frac{|M_z|\,w}{2\,I_z} + \frac{|M_y|\,h}{2\,I_y} + \frac{|F_x|}{A} \tag{2.147}$$

Since k_c only applies to the inner edge of the beam, it is possible that F_x *reduces* stresses there, which would lead to a slight von Mises stress overestimation. The non-planar bending term should also have a stress concentration factor, but Young and Budynas do not provide one, so Eq. (2.147) will remain somewhat inaccurate for M_y-loading. Perhaps future versions of CoMeT will include the appropriate factor.

For maximum torsional shear, the rectangular beam expression from Eq. (2.137) is used. This is a significant approximation, since Poisson effects induce anticlastic curvature, leading to warping of the neutral axis and highly non-intuitive, complex stress fields.

Eq. (2.147) is numerically maximized over the length of the beam to give the maximum stress. In order to do so, loading must be transformed to a coordinate frame tangent to the current cross section. This is done via matrix transformations. The angle α gives the location of the given cross section, as measured from the apex of the curved beam, and θ is shown in Fig. 2-11.

$$r_x = \cos\theta - \cos\alpha \tag{2.148}$$

$$r_y = \sin\theta - \sin\alpha \tag{2.149}$$

$$\mathbf{R}_{0/1} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.150}$$

$$\mathbf{C}(\mathbf{r}_1) = \begin{bmatrix} 0 & 0 & r_y \\ 0 & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{2.151}$$

$$\mathbf{T}_{0/1} = \begin{bmatrix} \mathbf{R}_{0/1} & 0 \\ \mathbf{R}_{0/1}\mathbf{C}(\mathbf{r}_1) & \mathbf{R}_{0/1} \end{bmatrix} \tag{2.152}$$

$$\mathbf{F}_\alpha = \mathbf{T}_{0/1}\,\mathbf{F}_\ell \tag{2.153}$$

Note that since r_x and r_y are expressed in the ℓ-frame, Eq. (2.152) first translates the

beam loads to the centerpoint of the current frame, *then* rotates through an angle α.

Given $\mathbf{F}_\alpha$, the forces and moments required for Eqns (2.147) and (2.137) may be extracted, Eq. (2.135) evaluated, and the maximum stress found.

2.5.4 Tapered Beams

Tapered beams are modelled as the continuum limit of varying rectangular beams in series, and the maximum stress is found consistent with that approximation. Although it it quite likely that the highest stress is at one of the two beam ends if parameters are chosen at random, a properly optimized tapered beam will avoid precisely that. It is therefore necessary to numerically maximize the highest von Mises stress in a cross section given by Eqns (2.135), (2.136), and (2.137) over the length of the beam. Since the surface is sloped and material at the surface must be in plane stress, there will be shear stresses in a tapered beam under every type of loading. These generally have minor effects, so long as surface slope does not exceed 15°. The torsionally induced shear stress suffers most under the approximation, since the two shear fields interact one-to-one. Subsection 5.1.3 discusses the tapered beam discrepancies in detail.

2.6 Instant Screw Axis Representation

Any planar rigid body motion can be represented as pure rotation about a special point, known as the instant center. In three dimensions, motion can be captured by an instant screw axis [16, p. 114]. There is thus translation along the screw axis and rotation around the axis,

$$\delta_N = \delta_{IC} + \theta_{IC} \times \mathbf{R}_{IC-N} \tag{2.154}$$

where N refers to an arbitrary point, or node, on said body. The situation is depicted in Figure 2-16. The rate of translation is not independent of the rate of rotation; it is related by the imaginary 'pitch' of the screw axis. If the 'pitch' happens to be

zero, motion is again planar; and if the plane of motion is known (as given by the orientation of the screw axis), the axis projects to a point in that plane, and may be called an instant center.

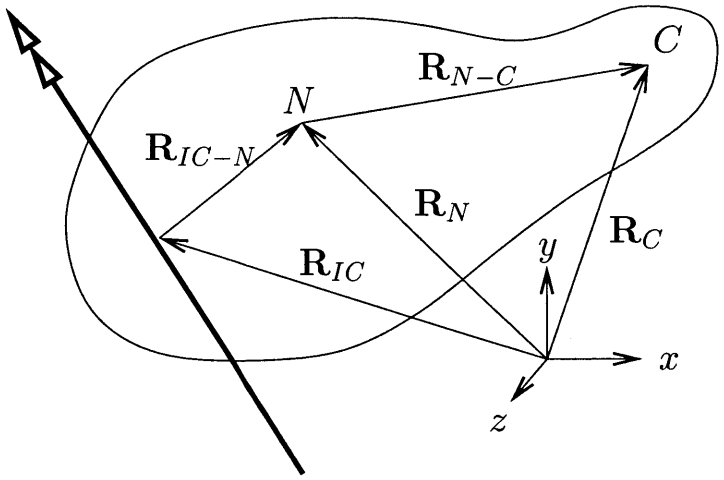


Figure 2-16: Motion about a screw axis

In any case, we are interested in finding $\mathbf{R}_{IC}$ and θ_{IC}, as these fully define the screw axis, for a given motion at some node N. Note that $\mathbf{R}_{IC-N}$ points from node N to the closest point on the axis, and we can later find $\mathbf{R}_{IC}$ using Eq. (2.155). In addition, we would like to find the motion of a specified centroid C that is rigidly attached to N.

$$\mathbf{R}_{IC} = -\mathbf{R}_{IC-N} + \mathbf{R}_N \tag{2.155}$$

Since the instant center, node N, and centroid C are on the same body, they must experience identical rotations:

$$\theta_N = \theta_{IC} = \theta_C \tag{2.156}$$

To determine $\mathbf{R}_{IC-N}$, we are interested in the rotation term in Eq. (2.154). In the plane of θ_N and δ_N, as pictured in Figure 2-17, we may separate displacements perpendicular and parallel to the rotation vector.

$$\delta_{N_\perp} = \delta_N - \delta_{N_{//}} \tag{2.157}$$

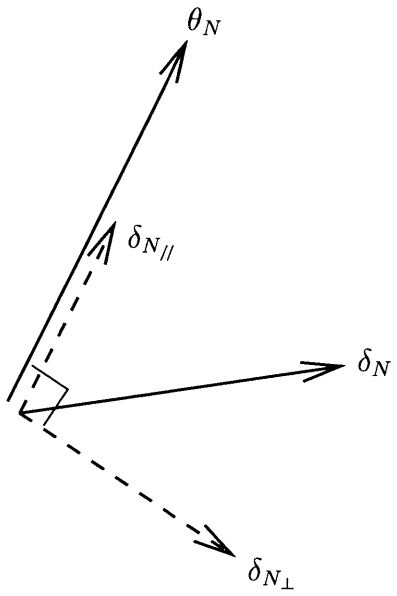


Figure 2-17: Decomposition of displacement along and orthogonal to the rotation vector

$$\delta_{N_\perp} = \delta_N - \delta_N \cdot \hat{\theta}_N \tag{2.158}$$

By definition, $\delta_{N_\perp}$ can only stem from the rotation term of Eq. (2.154):

$$\delta_{N_\perp} = \theta_N \times \mathbf{R}_{IC-N} \tag{2.159}$$

Since the cross product operation is not one-to-one, Eq. (2.159) can not be inverted *per se*. However, by specifying that $\mathbf{R}_{IC-N}$ point to the closest point on the instant screw axis, which may be done by declaring it perpendicular, $\mathbf{R}_{IC-N}$ becomes uniquely defined.

$$|\mathbf{R}_{IC-N}| = \frac{|\delta_{N_\perp}|}{|\theta_N|} \tag{2.160}$$

$$\widehat{\mathbf{R}}_{IC-N} = \hat{\theta}_N \times \hat{\delta}_{N_\perp} = \frac{\theta_N \times \delta_{N_\perp}}{|\theta_N||\delta_{N_\perp}|} \tag{2.161}$$

By multiplying Eqns (2.160) and (2.161),

$$\mathbf{R}_{IC-N} = \frac{\theta_N \times \delta_{N_\perp}}{|\,\theta_N|^2} \tag{2.162}$$

Using Eq. (2.158), the numerator may be simplified:

$$\theta_N \times \delta_{N_\perp} = \theta_N \times \left(\delta_N - \delta_N \cdot \hat{\theta}_N\right) = \theta_N \times \delta_N \tag{2.163}$$

With the help of the definition of vector magnitude, we can now specify $\mathbf{R}_{IC-N}$ in terms of known quantities:

$$\mathbf{R}_{IC-N} = \frac{\theta_N \times \delta_N}{\theta_N \cdot \theta_N} \tag{2.164}$$

Equation (2.164) gives a vector pointing from the instant screw axis to node N. In practice, the global coordinates of the screw axis are most useful, as shown in Figure 2-16. Combining Eq. (2.155) with Eq. (2.164), Eq. (2.165) finally gives $\mathbf{R}_{IC}$:

$$\mathbf{R}_{IC} = \frac{\delta_N \times \theta_N}{\theta_N \cdot \theta_N} + \mathbf{R}_N \tag{2.165}$$

In order to find motion at centroid C, instant screw axis principles could be invoked, but are not necessary. From elementary vector kinematics, it should be obvious that

$$\delta_C = \delta_N + \theta_N \times (\mathbf{R}_C - \mathbf{R}_N) \tag{2.166}$$

The implementation of relations derived in this section is detailed in Section 4.4.

2.7 Actuation Matrices

CoMeT computes several matrices pertaining to actuation. These comprise $\mathbf{K_A}$, $\mathbf{C_A}$,

$\mathbf{F_{A,\,max}}$, $\mathbf{x_{A,\,max}}$, $\mathbf{S_x}$, $\mathbf{S_F}$, $\mathbf{S_{x,\,F0}}$, $\mathbf{S_{F,\,x0}}$, $\mathbf{S_x^{-1}}$, and $\mathbf{S_F^{-1}}$.

$\mathbf{K_A(i,j)}$ contains the forces measured at actuator i when actuator j is displaced by a unit amount.

$$\mathbf{F_{A_i}} = \mathbf{K_A(i,j)}\ \ \mathbf{x_{A_j}} \tag{2.167}$$

In a completely decoupled system, $\mathbf{K_A}$ is diagonal, and no actuator induces forces or motion in any other actuator. Compliant mechanisms are likely to have some cross coupling however, and these must be considered in accurate analysis. In any case, the diagonal of $\mathbf{K_A}$ lists the actuator self-stiffnesses (given zero displacements at other actuators) and is the most frequently utilized portion of the matrix. The cross terms are useful for error budgeting and other computations.

The actuation stiffness matrix is simply a collection of the appropriate portions of $\mathbf{K_G}$, projected in the direction of actuation.

The $\mathbf{C_A}$ matrix lists the actuator self- and cross-compliances. It is derived using Eq. (2.168):

$$\mathbf{C_A} = \mathbf{K_A}^{-1} \tag{2.168}$$

Alternatively, $\mathbf{C_A}$ could be extracted from $\mathbf{C_G}$.

The diagonal of $\mathbf{C_A}$ is again the most significant. In a completely decoupled system, element-wise inversion is equivalent to matrix inversion given by Eq. (2.168). In partially coupled systems, the difference may be used to assess the degree of coupling.

The maximum allowable actuator forces are given by $\mathbf{F_{A,\,max}}$. This column vector is found by scaling each load magnitude by the ratio of the highest stress it induces in any beam to the material yield stress. Since stresses are nonlinear, superposition does not apply to $\mathbf{F_{A,\,max}}$.

Often the actuator work volume is of greater concern. The column vector $\mathbf{x_{A,\,max}}$ finds the largest actuator input displacements that does not cause yielding, assuming all other actuators are free. It is, in MATLAB syntax,

```
xAmax = diag(CA).*FAmax
```
(2.169)

The remaining matrices relate centroid motion to actuation. $\mathbf{S_F}$ gives the global centroid displacement given unit actuation:

$$\begin{bmatrix} x \\ y \\ z \\ \theta_x \\ \theta_y \\ \theta_z \end{bmatrix}_{centroid} = \mathbf{S_F} \begin{bmatrix} F_{A_1} \\ F_{A_2} \\ . \\ . \\ . \\ F_{A_{NL}} \end{bmatrix} \tag{2.170}$$

Where NL is the number of loads or actuators defined for a structure. $\mathbf{S_F}$ has dimensions of compliance, and is extracted from $\mathbf{C_G}$ using the appropriate projections and transformations. In shorthand notation,

$$\mathbf{x_C} = \mathbf{S_F}\ \mathbf{F_A} \tag{2.171}$$

If actuators are displacement sources, $\mathbf{S_x}$ is needed to find centroid motions.

$$\mathbf{x_C} = \mathbf{S_x}\ \mathbf{x_A} \tag{2.172}$$

CoMeT calculates $\mathbf{S_x}$ using Eq. (2.173).

$$\mathbf{S_x} = \mathbf{S_F}\ \mathbf{K_A} \tag{2.173}$$

$\mathbf{S_x}$ can be interpreted as a matrix that allows one to apply displacement constraints to a structure. Displacement actuators are conceptually no different from displacement constraints, just as applied loads behave like force actuators. Each load thus creates one row in $\mathbf{S_x}$, which removes one degree of freedom. It is therefore possible to create arbitrary partial nodal constraints. The only limitation is that Sx will only

be immediately useful if all actuators are of the displacement type. A general mixed constraint solution may be implemented in future versions of CoMeT.

Sometimes actuators are displacement sources, but only one actuator is allowed to act at a time, a common scenario during calibration. In this case, $\mathbf{S_{x,F0}}$ gives the desired results:

$$\mathbf{x_C} = \mathbf{S_{x,F0}}\,\mathbf{x_A} \quad \text{| all but one actuator applying zero forces} \tag{2.174}$$

Please note that $\mathbf{S_{x,F0}}$ is computed for convenience only and does not permit linear combinations. If more than one displacement type actuator is used, the other actuators should be removed and $\mathbf{S_x}$ regenerated.[2]

To round out the list, $\mathbf{S_{F,x0}}$ follows below.

$$\mathbf{x_C} = \mathbf{S_{F,x0}}\,\mathbf{F_A} \quad \text{| all but one actuator applying zero displacements} \tag{2.175}$$

The $\mathbf{S_{F,x0}}$ matrix could be used to implement an arbitrary set of zero-displacement constraints, mixed with *one* non-zero force constraint.

Inverse kinematics are the design cornerstone to compliant positioning stages, such as the Hexflex®. They are required whenever centroid motion is known (desired), and actuator inputs must be found. CoMeT inverts $\mathbf{S_x}$ and $\mathbf{S_F}$ whenever these are square and invertible (that is, when the centroid DOF match NL, and the actuation configuration is not redundant). It even goes a step further and sees if there are DOF in actuator null space (i.e. no actuator causes motion along those directions), removes that row from $\mathbf{S_x}$ and $\mathbf{S_F}$, and attempts to invert again. If CoMeT still does not succeed, but the user desires inverse kinematics, he or she must modify the

[2]If a given loading scenario happens to not cause motion along an actuator's direction of actuation, then whether a zero displacement or zero force constraint is applied there, the solution will remain the same. $\mathbf{S_x}$ can thus be used even if actuators are not physically present to restrict motion.

sensitivity matrices manually to obtain results.

$$\mathbf{F_A} = \mathbf{S_F}^{-1}\,\mathbf{x_C} \tag{2.176}$$

$$\mathbf{x_A} = \mathbf{S_x}^{-1}\,\mathbf{x_C} \tag{2.177}$$

$\mathbf{S_F}^{-1}$ and $\mathbf{S_x}^{-1}$ are stored in the global variables `SFinv` and `Sxinv`, respectively.

Chapter 3

Advanced Analysis

Advanced analysis modules include the optimizer, the motion diagnosis module, and the sensitivity study. The analyzer simply allows graphical probing of the variables defined in Chapter 2 and deserves no further elaboration.

3.1 The Optimizer

The optimizer, a routine that loops around `flex.m` and records the effects of varying some parameter, harmonizes with CoMeT's modular structure and computational efficiency. A screenshot of the optimizer is given in Fig. 3-1. In the lower left corner, a block of "pseudo-code" illustrates the optimization program flow. The innermost loop is controlled by MATLAB's `fminbnd` command, which uses a "Golden Section search and parabolic interpolation" [3] to satisfy a minimization constraint, but the algorithm need not be understood for ordinary use. The two outer loops are simple "brute force" number crunching cycles. The optimizer theory being mentioned, it is crucial to learn the program flow logic and realize the possibilities that the optimizer unfolds.

The optimizer is a flexible template that permits an endless variety of studies. A limited set of pre-conceived studies can be implemented via drop-down menus, such

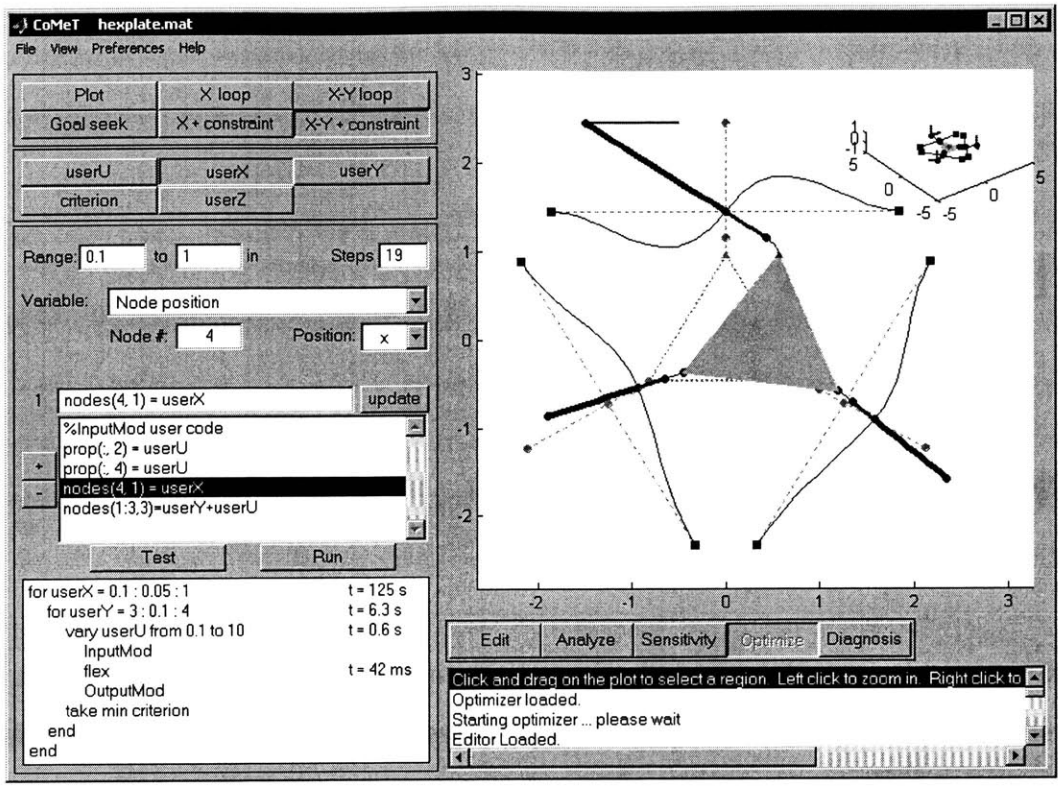


Figure 3-1: Optimizer screenshot including flow schematic.

as altering a node position shown in the figure. These accelerate the most common types of optimizations, but also serve as a built-in tutorial to users wishing to learn proper syntax. The user retains complete control over his optimization, and may manually alter the current study at any time.

The optimizer derives its versatility from MATLAB's `eval` command, which reads a string (text) variable and executes it as if it were hard coded. The powerful command thus permits complete output variable flexibility, multi-line computations[1], and even external script calls, should the user desire to save a more complicated study.

Discussion can proceed no further without the introduction of a few MATLAB variables, defined in Table 3.1. The names have been chosen systematically to facilitate quick memorization. Table 3.1 should be read in the order in which items

[1]To be precise, CoMeT stores user code in a cell vector whose elements are strings. The `eval` command is applied to each element.

are listed.

Table 3.1: **Key optimizer variables**

`InputMod`	Executed *before* each call to `flex.m`. This block of user code, accessible through the "Input" menu, alters variables that affect the current mechanism deformation response, such as `nodes` (node positions) and `prop` (beam properties).
`OutputMod`	Executed *after* each `flex` cycle. This second block of user code, accessible through the "Output" menu, defines user output quantities. These may now depend on any flex output variable, such as `Sx`, `RIC`, or `KA` (defined in Sec. 2.7).
`userX`	User input variable that varies through a user-specified range. Assignments should be made in `InputMod` and flow *from* `userX` *to* one of the core `flex` variables. Example: `prop(:,4) = userX` changes every beam height. Note that changing the final height of tapered beams would require another line, such as `prop(:,6) = userX`.
`userZ`	User Output variable used for plotting. The extrema of `userZ` and their corresponding `userX` and `userY` values are automatically computed. The assignment should be written in the `OutputMod` block and flow *from* flex output variables *to* `userZ`. Example: `userZ = max(sigma_max)` records the maximum stress in any beam, under any load.
`userY`	Used in addition to `userX`, in a similar way, for two-variable input variations.

`userU`	Independent variable of the minimization constraint loop. It is restricted to a certain range, but CoMeT does not step through the range linearly. Instead, MATLAB minimizes the number of iterations necessary to find the `userU` that minimizes `criterion`. Assignments should proceed as they do for `userX`.
`criterion`	Dependent variable to be minimized through `userU`. Equality constraints may be implemented using a square difference. Example 1: `criterion = TotalVolume` minimizes the total volume. Example 2: `criterion = (KA(1,1)-KA(4,4))^2` constrains actuators 1 and 4 to have the same self-stiffness.[2] Note that this formulation guarantees a solution, but the constraint is only met when `criterion` is approximately zero. Also, the convergence tolerance now applies to the square of the difference and is comparatively less stringent. The actual linear tolerance on $\mathbf{K_A}$ is the square root of the user-set tolerance parameter.

The CoMeT optimizer is itself modular. Various features can be mixed and matched as needed. The features are:

Goal seek - The simplest, fastest type of search. The problem can be stated as finding a special value of `userU` that minimizes criterion. The old $\texttt{userU}_o$ is optionally replaced by the optimum `userU`*. `userU`* may only be a local solution, but the user presumably knows intuitively that the solution is unique (or symmetric).

X-loop - More extensive single-variable optimization. Here, `userX` is varied evenly over a range and the `userZ` response recorded and plotted. Since the form of

[2] $\mathbf{K_A}$ and the term self-stiffness are defined in Sec. 2.7.

the entire `userZ` function is now available, multiple solutions, critical points, and output-input sensitivity (the slope of the curve) may be found.

XY-loop - Two-variable optimization where `userY` is varied in addition to `userX`. The key is that `userY` is now nested inside the `userX` loop, so `userZ` becomes a function of two variables. `userZ` can be represented as a surface plot, or "slices" may be taken through the `userZ` surface by imposing a linear constraint on `userX` and `userY`.

Multiple quantity tracking - In addition to the primary variable `userZ`, some quantities may be of secondary interest. These may be computed at the end of each loop and a second curve added to the `userZ` plot.

X-loop or XY-loop subject to constraints - This is essentially a combination of goal seek with a single or double variable loop. For each iteration, `userU` is varied to minimize `criterion`, and the constrained Z response to X and Y determined. Since this type of optimization study involves three nested loops, it takes the most time.

3.1.1 Optimizer Suggestions and Caveats

Although the optimizer theory is succinct, its application is one of the most challenging aspects of CoMeT. To assist the reader, an elaborate optimization example is given in Sec. 4.3.

The `userU-criterion` loop is intended to solve constraints on `flex` outputs, but it may be used on inputs just as well. The only drawback is an unnecessary calculation expense. For problems of medium difficulty, it may still be many times faster to use a CoMeT goal seek than to perform a hand calculation. This applies most notably to analytic point geometry. For example, a line's intersection with the midplane between two points could be found by varying a line-tracing point parametrically in `userU`, and minimizing the difference in distance to the two other points that define the mid-plane.

On the other hand, input constraints are often easily solved by hand, in which case computation time is immensely reduced.[3] Since CoMeT allows unlimited input modification lines, an arbitrary number of input constraints can be added with little computation penalty. Keep in mind that the user need not write a constraint in one explicit line; he can effect the constraint implicitly, while making unrestricted use of MATLAB's powerful computational tools.

In elaborate optimizations performed on complex structures, computation time may become considerable, and topological simplifications may be warranted. Section 3.3 explains how the sensitivity study may be used to identify beams that minimally affect a design. If these are replaced by plates or removed, many excess calculations may be avoided. The most time-intensive elements are, in order of decreasing significance, doubly tapered, curved, and simply tapered beams.

Long studies can be saved in separate `m`-files, with obvious advantages. To call a script from the user code block, simply type the name of the file on a separate line. Ensure that the file is in MATLAB's search path, or include the full file name. Functions may be called using the usual syntax.

3.2 Motion Diagnosis

The motion diagnosis tool presents two main concepts, $\eta - T$ decomposition and strain energy. The $\eta - T$ decomposition is an original contribution by the author that can be used to determine how efficiently two points of a mechanism are coupled. Strain energy shows which parts of a a mechanism are the most "active". Together, they provide powerful insights into compliant mechanism functionality.

[3] On a typical system, the time saved by eliminating a nested loop is 90-95% of computational duration.

3.2.1 $\eta - T$ Decomposition

The $\eta - T$ decomposition is the compliant analog of rigid body transmission ratio analysis. Since compliant mechanism behavior is more complicated than that of its traditional counterparts, $\eta - T$ analysis is correspondingly more involved. However, given an understanding of transmission ratios and the concept of efficiency, $\eta - T$ should be a natural extension to engineering knowledge.

In this subsection, a short review of traditional analysis is presented, with emphasis on how it motivates the $\eta - T$ approach. Next, mathematical relations and properties involving η and T are derived. The subsection concludes with a discussion of how $\eta - T$ is applied to mechanism design.

In kinematic devices **in which compliance is insignificant relative to other motions**, transmission ratios may readily be defined as follows:

$$T = \frac{x_2}{x_1} \tag{3.1}$$

Here, 1 is the point of actuation and 2 the point at which work is extracted. Eq. (3.1) implies two relationships:

$$x_2 = T\, x_1 \tag{3.2}$$

$$x_1 = \frac{1}{T}\, x_2 \tag{3.3}$$

Equations (3.2) and (3.3) must be true for all x, and a single T.

Furthermore, if a system is conservative,

$$F_1\, x_1 = F_2\, x_2 \tag{3.4}$$

Eq. (3.4) may be combined with Eqns (3.2) and (3.3) to yield a relationship between

forces:

$$F_2 = \frac{1}{T} F_1 \tag{3.5}$$

$$F_1 = T F_2 \tag{3.6}$$

Again, T is one and the same for a given mechanism.

In compliant mechanisms, none of the previous relations are consistent. Although we will assume that the compliant mechanism is conservative, and energy only enters or leaves the system at points 1 or 2, energy will be stored in the mechanism, and Eq. (3.4) will no longer hold. Equations have to be written carefully with qualifying subscripts.

$$x_2 = H_F \, x_1 \Big|_{F_2=0} \tag{3.7}$$

$$x_1 = H_B \, x_2 \Big|_{F_1=0} \tag{3.8}$$

Here, H is used to denote a geometric advantage (also known as displacement transfer function), subscript F to indicate "forward" flow, subscript B for "backward" flow, and the slash subscript to remind us which external force is zero (and hence which point is forced).

If Eq. (3.7) holds, x_1 is *at most* as much as Eq. (3.9) indicates. If it were more, x_2 could be tied to x_1 of a replica of the mechanism, and energy created.

$$x_1 \leq \frac{1}{H_F} \, x_2 \Big|_{F_1=0} \tag{3.9}$$

Unfortunately, an inequality is a fairly weak statement. Instead, let us decompose H into a transmission factor T_{X} that is retained whether 1 or 2 is actuated, and an efficiency factor η_{X} that is the same in both cases. The subscript x is used to indicate

that a term is derived using displacement tests. In mathematical terms, we desire

$$x_2 = \eta_X T_X x_1 \Big|_{F_2=0} \tag{3.10}$$

$$x_1 = \eta_X \frac{1}{T_X} x_2 \Big|_{F_1=0} \tag{3.11}$$

Eqns (3.10) and (3.11) are the context in which η and T are best understood. After some minor mathematical manipulation that the reader is encouraged to recreate, η_X and T_X may be isolated.

$$\eta_X = \sqrt{H_B H_F} \tag{3.12}$$

$$T_X = \sqrt{\frac{H_B}{H_F}} \tag{3.13}$$

Given Eq. (3.9), η_x cannot exceed unity. Furthermore, both η_x and T_x must be nonnegative.

So far, displacements have been used to calculate η and T. A similar derivation may be based on forces. Rather than recording the displacement at a point when no external force is applied, we may may record the force required to counteract any motion.

$$F_2 = G_F F_1 \Big|_{x_2=0} \tag{3.14}$$

$$F_1 = G_B F_2 \Big|_{x_1=0} \tag{3.15}$$

G is used to denote a force advantage, or force transfer function. All other notation is analogous. Using similar logic, we wish to decompose forces as follows:

$$F_2 = \eta_F T_F F_1 \Big|_{F_2=0} \tag{3.16}$$

$$F_1 = \eta_F \frac{1}{T_F} F_2 \Big|_{F_1=0} \tag{3.17}$$

(3.18)

Again, through after some manipulation, we find

$$\eta_F = \sqrt{G_F G_B} \tag{3.19}$$

$$T_F = \sqrt{\frac{G_F}{G_B}} \tag{3.20}$$

We will now prove the surprising result that T_x and T_F, and likewise η_x and η_F, are always equal! For any conservative quasi-static linearly elastic structure, we may take two arbitrary directions 1 and 2, and construct a stiffness matrix involving only displacements and forces in these directions:

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3.21}$$

Note that this equation differs significantly from Eq. (2.30). Displacements and forces at each point are projected in the specified directions, so all entries are scalar. Furthermore, points 1 and 2 may be anywhere in a structure, and connected by any combination of linearly elastic elements.

An analogous compliance matrix may be defined as well:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \tag{3.22}$$

With the help of Eqns (3.7), (3.8), (3.14), and (3.15), we may derive that

$$H_F = \frac{C_{21}}{C_{11}} \tag{3.23}$$

$$H_B = \frac{C_{12}}{C_{22}} \tag{3.24}$$

$$G_F = \frac{K_{12}}{K_{11}} \tag{3.25}$$

$$G_B = \frac{K_{21}}{K_{22}} \tag{3.26}$$

The compliance matrix must be the inverse of the stiffness matrix. Inverting and setting like terms equal produces the results given below.

$$H_F = -G_B \tag{3.27}$$

$$H_B = -G_F \tag{3.28}$$

Finally, combining (3.19), (3.20), (3.27), and (3.28), we establish the heralded result:

$$\eta_{\mathrm{X}} = \eta_F \equiv \eta \tag{3.29}$$

$$T_{\mathrm{X}} = T_F \equiv T \tag{3.30}$$

If a disturbance partially blocks a mechanism, weighted superposition of the no load and no displacement cases will always yield the same decomposition.

Eq. (3.30) shows that T is an invariant transmission ratio for a given structure, load, and motion point. It quantifies the amount of reduction achieved by reversible mechanical means.

Meanwhile, η indicates how well motions at two points are correlated. It is an information loss factor, a measure of how well two points communicate. The more η approaches unity, the more the operative portion of a flexure behaves like a rigid mechanism.

The η-T decomposition ought to be as useful for compliant mechanisms as transmission ratio analysis is for rigid devices. Since both quantities are dimensionless, they may be used to compare entirely different designs.

A large η can be desirable for several reasons. For small disturbances, it gives the

fraction of disturbance energy that is taken by an actuator. The remaining energy must be stored elsewhere in the system. Since actuators are usually much stiffer than the system they are used in, whether mechanically or virtually via internal feedback, a larger η will lead to less uncontrolled deformation.

In compliant grippers, such as those developed by Kotah et al [4] and Sigmund [15], high η values would indicate good input-output stiffness, since disturbance displacements would attempt to move the actuation site, which is presumably rigid, and generate large restoring forces.

Sometimes, low η values are unavoidable and even desirable. It is usually easier to achieve large motion de-amplification through η than through T. Disturbances at the input (which are indistinguishable from intentional inputs) will always be reduced by η and T alike, and if inputs dominate system uncertainty, the loss in output disturbance rejection may be irrelevant.

No matter what the mechanism, its engineer should know in what range η and T lie. If η is small, stiffening the system should best reduce the effects of disturbances. If η is large, finding stiffer actuation or increasing η some more may be the more cost efficient strategy.

Since η-T decomposition is a new concept, it is likely to have other, undiscovered significance to precision engineering. At the moment, a sound framework for mechanism analysis has been established. Future work should reveal further implications and benefits of performing the analysis.

3.2.2 Strain Energy

The distribution of strain energy reveals which elements of a mechanism are most active. Elements that store energy are more likely to be critical to system performance. If creep is a concern, strain energy should be minimized. Whatever the functional requirements may be, CoMeT's motion diagnosis provides fast graphical insight into a mechanism's strain energy distribution.

Since motion is assumed linear, strain energy is exactly half the current force times

the current displacement. In scalar form:

$$W_{strain} = \int F \, \mathrm{d}x = \int Kx \, \mathrm{d}x = \tfrac{1}{2}Kx^2 = \tfrac{1}{2}Fx \tag{3.31}$$

Eq. (3.31) can be generalized to multiple DOF elements using the dot product.

$$W_{strain} = \tfrac{1}{2} \sum \mathbf{F_{beam}} \cdot \mathbf{x_{beam}} \tag{3.32}$$

Here, $\mathbf{x_{beam}}$ and $\mathbf{F_{beam}}$ are [12x1] matrices, since each beam has twelve degrees of freedom. Since CoMeT computes $\mathbf{F_{beam}}$ as a [6x1] matrix of internal beam forces, Eq. (2.31) must be invoked.

$$W_{strain} = \tfrac{1}{2} \sum \mathbf{F_{beam}} \cdot \left(\mathbf{x_2} - \mathbf{T}_l{}^T \, \mathbf{x_1} \right) \tag{3.33}$$

As usual, $\mathbf{x_2}$ contains generalized displacements at the second beam node, and $\mathbf{x_1}$ at the first, both expressed in the beam's local frame.

The sum of the element strain energies should equal the work done by the actuators.[4] Under a variety of test cases, CoMeT found the two values to differ by one part in 10^{12}, lending further credibility to the accuracy of its calculations.

3.3 Sensitivity Analysis

In CoMeT's sensitivity analysis, a completed compliant mechanism is tested for sensitivity to variations in its geometry. The quantity whose sensitivity is measured, known as the criterion, may be anything the user specifies. All primary beam properties (h, w, a_o, a_f, b_o, b_f, and R, as applicable by beam type) and every node position (r_x, r_y, and r_z) are perturbed by a small amount, and the criterion change

[4] In the absence of loads or disturbances, that is. A load behaves like an actuator doing negative work, extracting energy from the system.

is recorded. The structure plot is then color coded, with the most sensitive beams or nodes appearing red, completely insensitive elements appearing black, and everything in between interpolated accordingly.[5] Beams and nodes can then be probed with a graphical pointing device, and exact sensitivities are displayed in a menu.

3.3.1 Why Perform a Sensitivity Analysis?

In any design, it is important to understand which parameters most affect system performance. This is essential for manufacturing, optimization, and modelling. A sensitivity study can quickly reveal which geometry is most critical to a design, but usually requires many iterations. Even rough hand tolerance calculations usually take time, and engineers have tended to resort to intuition and experience in the past. CoMeT's systematic, quick, and easy sensitivity analysis should encourage more deterministic approaches.

Whether a compliant mechanism is produced in single or mass quantities, it cannot be produced cost effectively without an understanding of the required tolerances. A single unit may well tolerate defects in some areas, but require fixing or re-building if others are faulty. Sometimes sensitivities are "so high" that a mechanism can only be made from special precision materials, such as stress-relieved metal. Similarly, manufacturing methodology may prove inadequate for a given cause. If tolerances are established through trial and error alone, much time and effort may be lost in the process. For mass production, CoMeT may be used to provide initial tolerance estimates, which can later be refined empirically.

A sophisticated compliant mechanism is likely to have dozens of elements and nodes, making optimization appear a daunting task. The sensitivity study may provide a starting point for in-depth investigations. If it reveals performance to be very sensitive to one parameter, optimizing that parameter is more likely to yield large improvements. Sometimes a parameter does not affect performance as much as expected; in this case, the sensitivity study may shed light on which part of a

[5] Advanced users may alter the color scheme if they wish.

mechanism is not functioning properly. Finally, low sensitivity may be a goal in its own right, since a robust design is more likely to succeed in the "real world". In this case, the sensitivity chart may identify areas that require conceptual redesign.

Lastly, an understanding of mechanism sensitivity is crucial for proper analytic modelling. Often, a rough analytic "stick figure" model of a more complicated mechanism is desired. CoMeT cannot derive analytic equations, but it can help indicate which elements require special attention and which may be treated with less care. Combined with the motion diagnosis strain energy plot, the sensitivity analysis can give an engineer the information needed to make educated decisions about which elements and parameters to include in a model.

3.3.2 Sensitivity Analysis Theory

A dimensionless sensitivity that resembles the derivative may be defined as follows:

$$\text{dimensionless sensitivity} = \frac{Z(X + \epsilon X) - Z(X)}{\epsilon Z} \tag{3.34}$$

Here, Z is the criterion of the user's choosing, such as maximum stress, total volume, stiffness in a specified direction, or a transmission ratio. X is the currently varied parameter, such as beam width, curved beam radius, or node position. The variable ϵ is a small parameter, set by default to 0.01 (one percent).

The dimensionless aspect of Eq. (3.34) has several benefits. If sensitivity is unity, then a 1% parameter variation will lead to a 1% criterion variation, and so forth. In essence, sensitivity is the "variability gain" for a system. In addition, the actual value of ϵ used in computation, barring round-off error or large nonlinear change, will not affect the outcome. The sensitivity magnitude is therefore meaningful in a more general context that involves neither the geometric extent of the mechanism nor the particular ϵ used during computation.

Although Eq. (3.34) has the aforementioned benefits, there is an important limitation. While in many situations beam parameters are likely to vary in proportion

to their original size, node position sensitivities have dubious significance. Unless the origin has special importance in manufacturing, as by being the location of the inlet port of an injection-molded mechanism or the fixturing center in a rig, there is usually no reason to weigh sensitivity inversely with distance from the origin, which is what happens when absolute coordinates are multiplied by a fixed percentage. As long as this facet is born in mind, the non-dimensional sensitivity may prove useful, and CoMeT will analyze nodes in accordance with Eq. (3.34) if nondimensional sensitivity is selected.

A second type of sensitivity requires a fixed dimensional step, but is meaningful for all geometric parameters and also the more realistic in many manufacturing settings.

$$\text{dimensionally driven sensitivity} = \frac{Z(X+\gamma) - Z(X)}{Z} \tag{3.35}$$

Although the sensitivity expressed in Eq. (3.35) is still a dimensionless quantity (since it is normalized by Z), it now depends on the quantity γ. Since only geometric parameters are varied, and γ must have the same units as X, γ takes on the current length unit.

A fixed geometric variation amount is quite common in manufacturing. For example, a waterjet cutter and a milling machine each have absolute tolerances on the order of 0.001 inches. The dimensionally driven sensitivity study, if the process accuracy is used for computation, therefore allows immediate assessment of process capability adequacy. If γ is smaller than the shortest node separation or beam parameter, CoMeT will not crash, but negative geometry might result in command line error messages and incorrect sensitivities.

3.3.3 Sensitivity Analysis Caveats

The sensitivity study is a powerful tool that quickly generates valuable results. Nevertheless, the algorithm is crude, and human judgement needs to supersede numerical results. If parameters are expected to vary substantially from the analyzed

configuration, sensitivities may not be linear, and results may be misleading.

Eqns. (3.34) and (3.35) are singular when the criterion is initially zero, so CoMeT automatically modifies the calculations, removing the denominator in either case. The result is that the Z-variation now has the dimensions of Z, whatever they may be, and requires careful interpretation.

Nodal positions are varied along the global xyz-coordinate system only, so sensitive directions that do not align with the coordinate frame could be missed. Relative sensitivity may be inaccurate by a factor of two (and more in certain rare configurations), since there is no guarantee that the *most* sensitive direction is ever tested. The premise is that if there is a sensitive direction, its sensitivity will project onto at least one of the orthogonal bases and alert the designer.

On a similar note, the number of tested scenarios is limited to those listed in subsection 3.3.2. When a node moves, it stretches or shortens all the beams connected to it, so beam length is never varied in isolation. It may well be that two strong effects happen to cancel each other, and low sensitivity is recorded. Perhaps the position of the other beam nodes or the geometric beam properties would still reveal the sensitive area, but there is no guarantee that this will happen.

Finally, but certainly not least, the proper criteria must be considered. CoMeT cannot guess which quantities matter to the designer, and considerable wisdom is required to identify and formulate the right ones. If a designer's intuition warns him that a parameter is important, but sensitivity studies do not immediately back up his opinion, it is better to err on the side of caution and examine it closely.

Chapter 4

The Program

4.1 Core Variables and the Command Line

As explained in Sec. 1.5, CoMeT may be driven via the command line or the GUI. Although the GUI is more intuitive, the command line remains the fastest and most flexible way to enter data.[1] To create CoMeT files on the command line, the user need only learn the syntax of the core variables listed in Table 4.1.

Table 4.1 employs a few conventions that need be understood. The "dimensions" column contains the size of the current variable, rows by columns enclosed in square brackets. Matrix sizes that depend on the particular structure are indicated by a short, capitalized variable (such as `N` or `Gr`). The unspoken implication is that each instance of an object is numbered by its place along the dimension of variable size. For example, `B`, the number of beams, is a row placeholder, so the nodes listed in row i of `beams` define the endpoints of beam number i. The `prop` matrix also contains `B` rows, so row i of `prop` holds beam i's properties.

Notice that nodes (and later, other objects) are referred to by number, not their

[1]The reader is encouraged to learn the use of MATLAB's colon (:) and square bracket ([...]) operators.
Example: `A(3:5,:)=[ ]` deletes rows 3, 4, and 5 of matrix A.

properties (coordinates). This gives them a unique logical identity, while ensuring that property data is stored in a single place. If a node is later moved (i.e. its coordinates changed), no changes need be made to beams, loads, grounds, centroids, and plates that reference that node.[2] Nodes are allowed to have identical coordinates.

Table 4.1: **Core variables**

Variable	Dimensions	Description
`nodes`	[N x 3]	Rows listing the coordinates of each node. The columns hold x, y, and z data, in that order.
`grounded`	[1 x Gr]	Row vector listing of grounded node numbers.
`beams`	[B x 2]	Rows listing the end nodes of each beam element.
`prop`	[B x 2]	Row listing the properties that define each beam. `prop` may be considered an extension of `beams`. For details, see below.
`loaded`	[NL x 7]	Rows listing the node that a load acts on, followed by a generalized force vector. In shorthand: [load node#, F_x, F_y, F_z, M_x, M_y, M_z].
`centroid`	[C x 4]	Rows listing the node that a centroid is "rigidly attached" to, followed by the centroid x, y, and z coordinates. All centroid data will be expressed in the centroid's coordinate frame. In shorthand: [centroid node#, x, y, z].
`plates`	[P x ?]	Rows listing the nodes in a given plate. Zeros are used as placeholders and ignored by CoMeT. The `plates` matrix expands to accommodate the largest plate. Zeros are necessary if plates contain different numbers of nodes.
`E`	scalar	Elastic or Young's modulus of the material.
`G`	scalar	Shear modulus of the material.
`rho`	scalar	Density of the material.

[2]Deleting a node is another matter. The GUI may prove more convenient for this purpose, as it checks and corrects all references.

Table 4.1: (continued)

Variable	Dimensions	Description
`sigmay`	scalar	Yield stress of the material used for actuator limitation calculations. If the user wishes to design to the endurance limit or the ultimate tensile stress, `sigmay` may be changed accordingly. Although safety factors *can* be built into `sigmay`, the user risks later confusion.

Note that numerical object properties are not associated with a particular unit system, so all values must be given in a consistent system of units. For example, if lengths are measured in inches, forces in lbf, and mass in slugs, then the elastic modulus and stresses must be expressed in PSI, moments in $in\text{-}lbf$, and density in slugs per cubic inch.

Finally, note that `E`, `G`, `rho`, and `sigmay` are scalars, so they must apply to the entire structure. In other words, the entire compliant structure must be either monolithic or bonded from parts of the same material.

Once all core variables are defined, they may be checked at the command line using

```
FindErrors
```

The `FindErrors` function detects syntax errors, corrects some problems automatically, and generates the appropriate error messages. It does not check if a structure is underconstrained, or thwart other high-level errors of that nature.

When all errors have been rectified, type

```
SaveCurrentFile
```

and enter the file name of your choice when prompted. This file may be loaded through the GUI, if so desired, but analysis may proceed at the command line as well. To do so, type

```
CoMeTize
```

and give the name of the file you wish to analyze. `CoMeTize` generates a long list of global variables that contain the processed data. For information on these variables, type

```
help flex
```

You should be able to recognize many of their names from the mechanics chapter of this thesis.

4.2 Node Renumbering

As explained in Sec. 2.1, grounded nodes must be listed last before the analysis begins. Furthermore, Subsection 2.4.5 explains why plate nodes must be listed in a particular order. The user is free to enter nodes in any order, and CoMeT automatically renumbers nodes appropriately. But although node numbering is automatic, it must be born in mind, since analyzed data will use the new numbering system. For example, an explicit node reference made in the advanced modules, such as `dX(4)`, may require manual updating when nodes are deleted, grounded, or added to a plate.

If the user wishes to renumber nodes himself, he may run

```
RenumberNodes
```

at the command line.

4.3 The Optimizer, Revisited

Although the optimizer can generate code for some common optimization studies, is it impossible to anticipate everything the user may wish to do. The beauty of open source MATLAB is that the possibilities are endless - if the user knows how to implement them. This section gives an extensive example that makes use of several programming "tricks" that should prove useful in any advanced study.

The goal of the exercise is to optimize a doubly tapered beam. The beam is in a known state of three dimensional loading. Mass must be minimized subject to a

planar stiffness of 1.1, a non-planar stiffness of 1.2, and stresses not exceeding the yield stress.[3] The variables a_o, b_o, a_f, and b_f are to be determined.

The first "trick" is to isolate the tapered beam from the rest of the structure and recreate the local loading conditions. Although this step is not essential, it simplifies the problem conceptually and computationally. The internal beam forces (in the frame of the beam) can be retrieved by probing it in the analyzer. Better yet, they can be read directly from `Fbeam`. To do so, determine the number of the tapered beam (say, 8), and type at the command line

```
global Fbeam loadmag
FbeamCombinedLoading=Fbeam*loadmag
index = (8*6)-5   %each beam has 6 DOF and takes up 6 rows
Fbeam8=FbeamCombinedLoading(index:index+5,1)
global PlanarLoad NonplanarLoad   %to be used later
PlanarLoad=[Fbeam8(1:2),[0 0 0],Fbeam8(6)]'
NonplanarLoad=Fbeam8 - PlanarForce
```

The tapered beam may be re-drawn, grounded on one end, and a load set applied to the other (using the load set tool on the load menu panel). Since the internal forces are expressed in the *second* beam node's frame, it is essential that the *first* node be grounded. Finally, a centroid should be attached to the second node.

The next issue to recognize is that there are four variables of choice, but CoMeT currently only allows three user variables. As discussed in Sec. 6.1, the more powerful `fgoalattain` function from the optimization toolbox could handle this problem much more easily in the future, but the geometric scaling technique discussed in the next paragraph is required at the current time, and remains a useful technique to be aware of.

The minimum mass constraint may be handled analytically using geometric scaling. If width and height are scaled (multiplied) by the same factor λ, stress will scale with $1/\lambda^3$, and stiffness with λ^4. In other words, stress is proportional to

[3]Stiffness units are irrelevant for the purpose of this exercise, but must be consistent with the currently active unit system.

volume under equal radial scaling (length staying constant!). Note that CoMeT's tapered beam theory ignores surface slope effects already, so this is an exact relation inside CoMeT, but a slight approximation for strongly tapered beams. So we may take one variable as given, say $a_o = 0.1$, and find volume by adding the following to the start of the `OutputMod` block:

```
lambda = (max(sigma_max)/sigmay)^(-1/3)
%keep lambda2 for later use
global lambda2
lambda2(useri,userj)=lambda
userZ = TotalMass/lambda^3
```

The InputMod block is relatively straightforward:

```
prop(3:6) = [0.1 userU userX userY]
```

Next, the criterion must be assigned. We could have picked any two out of planar compliance, non-planar compliance, and stress to optimize in the inner loop; since the two compliances have similar magnitudes, we chose these.

Although `fminbnd` only permits single variable optimization, it can be used to solve for two parameters. We can write a single criterion to be minimized that "captures" both compliance constraints as follows:

```
global PlanarLoad NonplanarLoad %retrieve the loads we found before
PlanarCompliance = CG*PlanarLoad    %this is a [6x1]
%we want the magnitude of linear (i.e. x,y,z) compliance
PC = sqrt(sum(PlanarCompliance(1:3).^2))
NonplanarCompliance = CG*NonplanarLoad
%this time, using CoMeT's convenient vecmag command:
NPC = vecmag(NonplanarCompliance(1:3))
%now we apply scaling
PC2 = PC * lambda^4
NPC2= NPC * lambda^4
criterion = (PC2-1.1)^2 + (NPC2 - 1.2)^2
```

As long as both constraints are obtainable, it should not matter that the two terms are weighed equally.

After checking the code and running the optimizer, we determine the solutions we were searching at the command line. Metaphorically speaking, the `criterion` surface should show a "river" of `criterion`$\doteq$0 solutions running through a canyon of near matches. We can get the function that describes this river with the commands

```
global criterion lambda2 userX userY userZ
[solnRow, solnCol]=find(criterion < 0.001)
```

Note that the tolerance constant, 0.001, is a function of the grid resolution and may need repeated guesses. Next we construct the mass relation (which now meets stiffness and stress constraints) and minimize it. Note that we are obtaining coordinate pairs in space that need not lie along a single function.

```
mass3 = mass2(solnRow,solnCol)
[minmass, pointindex] = min(mass3)
```

Now we reconstruct all the variables we need to give us the intermediate optimum b_o, a_f, and b_f.

```
usericrit=solnRow(pointindex(1))
userjcrit=solnCol(pointindex(2))
bocrit=userU(usericrit,userjcrit)
afcrit=userX(usericrit,userjcrit)
bfcrit=userY(usericrit,userjcrit)
```

These variables are not yet scaled. So at long last we find a_o, b_o, a_f, b_f. Recall that the initial "guess" for a_o was 0.1.

```
%recycle lambda
lambda=lambda2(usericrit,userjcrit)
ao=0.1*lambda
bo=bocrit*lambda
af=afcrit*lambda
bf=bfcrit*lambda
```

Although this example may take some time to understand fully, doing so will allow you to optimize just about anything.

4.4 Variable Types and Syntax Issues

CoMeT utilizes some variable types that the user may not be familiar with. These are three dimensional or paged matrices, cell arrays, and structures. Detailed help can be found on the MATLAB documentation website [3] under "Using MATLAB" ⇒ "Programming and Data Types".

Three dimensional matrices are similar to two dimensional ones, but require three indices. A given compliant mechanism will have sensitivity ($\mathbf{S_x}$, $\mathbf{S_F}$, etc.) matrices for every defined centroid. `RIC` has an instant center position for every loading case, again for each centroid. CoMeT saves these variables in page matrix form. In MATLAB, the i^{th} sensitivity (where i is the centroid number) may be extracted using

```
Sxi = Sx(:,:,i)
```
(4.1)

Page matrices make many operations fast and convenient. For example, to average four sensitivity matrices, we could type

```
Sxavg = sum(Sx,3)/4
```
(4.2)

where the three indicates summation along the third (page) dimension, and the four is an indication that as many centroids were defined.

Cell arrays are matrices holding potentially dissimilar element types. The sensitivity study generates the cell array `slistordered`, which contains sensitivity (sorted in descending order) in the first column, and a short description in the second. Since doubles and strings are dissimilar, a normal matrix could not do the same.

To access or concatenate cell array entries, use curly brackets ({ ... }) in place of

square or round ones, respectively. Everything else, including colon syntax, remains the same. Thus, to retrieve the third largest sensitivity, one would type

```
s3=slistordered{3,1}
```

Note that

```
s3=slistordered(3,1)
whos s3
```

reveals that s3 is a [1 x 1] cell array, not a double precision integer.

Finally, the CoMeT GUI makes extensive use of structures. A structure is a loose hierarchical grouping of variables or other (sub-) structures into fields, just as a directory tree (a.k.a. directory structure) may be used to group similar files. To access a particular structure field, type the name of the structure, a period, and the name of the field.

The sensitivity study creates a structure, sensitivity, that has several fields. The following chunks of code illustrate manipulations one might normally apply to a structure.

```
%load sensitivity into the workspace
global sensitivity
%show sensitivity's fields:
sensitivity
%MATLAB's response:
sensitivity =

  beamprops: [27x4 double]
  nodepos: [30x3 double]
  thickness: 0.1234

%to copy beamprop to a conventional matrix:
beampropmatrix=sensitivity.beamprop
%to read the sensitivity to beam 5 height:
beampropmatrix(5,2)
%alternatively, we can index the field directly:
sensitivity.beamprop(5,2)
```

```
%to overwrite a field, just use an assignment.
sensitivity.beamprop = 'gone'
%In this example, the old beamprop gets deleted and replaced
%by the scalar string "gone".
```

4.5 GUI Layout

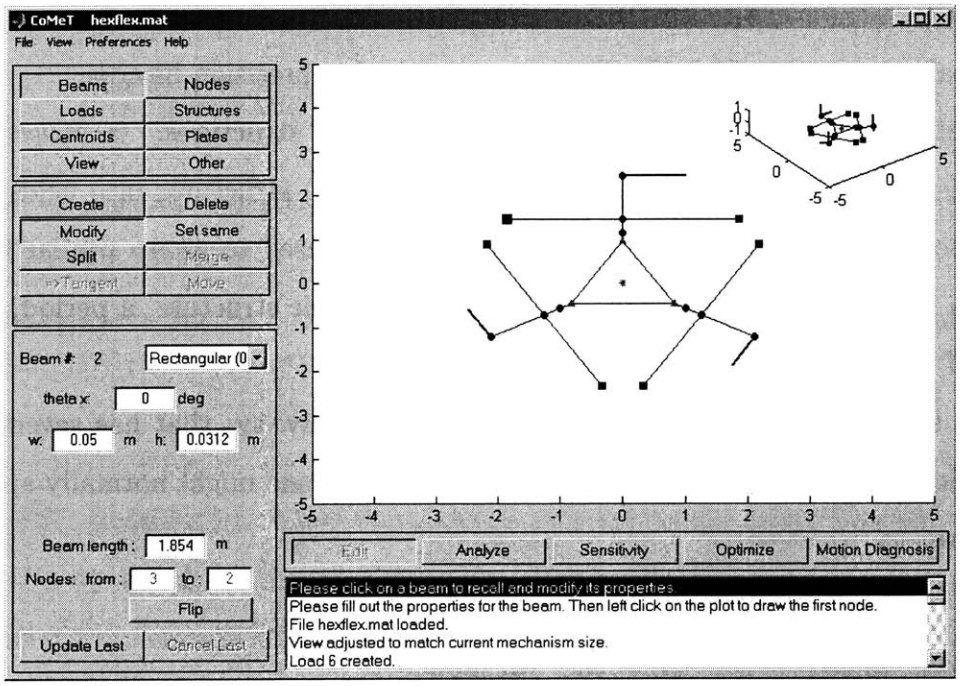


Figure 4-1: CoMeT editor layout

The CoMeT interface is highly structured and systemized. Once the interface logic is understood, the program may be learned and explored much faster. The objected oriented menu layout may prove an inspiration to other interface developers. Furthermore, the menu hierarchy is indicative of the GUI program structure. A basic knowledge of screen layout vocabulary is necessary for further discussion.

Fig. 4-1 shows a screenshot of the CoMeT GUI. The large plot showing a planar view of the Hexflex^{TM} is referred to as the plot area, or main plot. The small isometric version of the same structure in the upper right corner is called the thumbnail. Both

main and thumbnail views may be changed arbitrarily. Most sketching functions are only enabled in the main plot, but some, such as load deletion, are applicable to the thumbnail as well. This makes it possible to perform some functions immediately, when it would otherwise be necessary to change views beforehand.

Below the main plot is a five-button panel, the mode menu. Since the "Edit" button is depressed in Fig. 4-1, the editor is active. Below that is the message box, listing the last five[4] CoMeT messages, which contain helpful hints and alerts. The user may scroll to retrieve the last fifty[4] messages.

The mode menu is the pinnacle of a hierarchy:

Modes
↪ Objects
↪ Tools
↪ Properties and user I/O

Each menu expands a subordinate menu below it. Edit mode, for example, creates the edit objects menu, starting with "beams," seen on the upper left corner of Fig. 4-1. The objects correspond roughly to the core variables listed in Table 4.1. The view menu is accessible through every mode object menu, since it is useful in all contexts.

Object buttons create a panel of tool (or method) buttons below them. All tools are therefore grouped by the objects that they primarily act upon. In Fig. 4-1, the beam tools menu is visible. It contains all the "standard" methods — create, delete, modify, set same, split, and merge — that are always found in that order, if applicable to the current object.

The tool buttons have diverse effects. In most cases, they create property menus, such as the beam property menu shown, that allow the user to enter, read, and change values. Property menus are often "recycled" by different tools of that object. Some tools create highly specialized menus that follow no particular pattern. Yet others create no menus at all.

[4]The default. Customizable via `prefs`; see Sec. 4.8.

In addition to modifying the property menu area, tool buttons change an important variable, `status`, that determines how the GUI behaves. When the user clicks on the main plot, `status` decides if a node should be created, a load deleted, beam properties written to the beam properties menu, or whatever the case may be.

The GUI could be discussed indefinitely at this point, but the major CoMeT paradigms have all been stated. Trial and error will usually reveal what effect each button, checkbox, edit box, listbox, etc. has. In addition, most tools post instructional messages to the message box upon activation, and most buttons have a short, descriptive text associated with them that appears when the user's pointing device hovers over that button for a few seconds. When in doubt, save your work, then experiment with a newly discovered feature.

4.6 Structures and Patterning

The CoMeT GUI permits the user to form structures, which are, like the homonymous variable type discussed in Sec. 4.4, loose groupings of related objects. The list may include beams, nodes, loads, centroids, and plates. Structure tools are accessible through the structure button in the edit objects menu. Structures are not supported by `CoMeTize` or any other command line tools.

Once a structure is defined, commands may be applied at once to all the objects in that structure. For example, the "delete structure" command applies the "delete beam" method to each beam, the "delete load" method to each load, and so forth.

The most powerful aspect of structures are their patterning functions. Rotary, linear, and scaled versions of an entire structure may be created in a single step. These steps may be iterated a specified number of times to generate patterns. If a single iteration is requested and the first instance deleted, patterning functions become transformation tools. If a particular function is not found in the relevant object's tool menu, it may be defined in the context of a structure.

4.7 Deformed Beam Shape Display

CoMeT plots deformed beams as curves, a feature that must not be misunderstood. It is intended to make the display more intuitive, not to serve as the basis of a diagnosis. Table 4.2 summarizes the algorithm used to generate each beam shape, emphasizing which ones are approximate.

Table 4.2: **Deformed beam display algorithms**

Beam type	**Deformed shape display algorithm**
Rectangular	Deformed shape is the exact analytic linear solution.
Round	Deformed shape is exact.
Tapered	Deformed shape is a rectangular beam "curve fit" that may only match at the beam's ends. The less a beam is tapered, the more accurate the shape.
Curved	The planar deformed shape may be one of two varieties. CoMeT first assesses whether an elliptical fit is possible. If yes, it solves for the ellipse that matches planar end conditions. This ellipse is only intended to convey beam type and boundary slopes. If not, CoMeT resorts to a rectangular beam fit. Non-planar deformations are always represented by a projected rectangular beam fit. Both planar and non-planar rectangular beam fits are marginally realistic for slightly curved beams, but degenerate quickly with included angle.

In summary, only rectangular and round beams are exact. All other beam shapes may only be "trusted" at the two endpoints. If an accurate displacement at an intermediate point along a beam is desired, whether visually or numerically, the "split beam" function may be used to create a node there.

As is common in FEA programs, deformations are exaggerated by an amplification factor. CoMeT guesses the factor in an attempt to create a reasonable display, but

the algorithm is not perfect. Only nodal displacements are considered, so relatively large displacements at intermediate points, whether real or a consequence of curve fitting, may lead to unappealing results. The user may set the amplification scale numerically or by means of a logarithmic slider bar.

4.8 User Preferences

The CoMeT GUI is highly customizable. It is designed to gradually evolve as the user discovers new features, becoming increasingly refined. If the user likes a new set of preferences, he can save them to disk, and CoMeT will adopt them from that point onward. Otherwise, preferences revert during the next CoMeT session.

Preferences are saved to a structure called `prefs`, whose fields contain the preference information. For example, `prefs.FontSize` determines the font size, in points, used throughout the GUI. Preferences may be changed at any time, but only take effect when the object(s) they refer to is/are created. It may therefore be necessary to save preferences and re-start CoMeT to implement user preference changes completely.

Changes to preferences will be lost after the current session is closed unless saved to disk. To save, just select "Save Current Preferences" from the preference menu. Preferences are saved to `CoMeTPrefs.mat`. `CoMeTPrefs` contains a single variable, `prefs`, which is the aforementioned structure.

Preferences may be changed at the command line. To do so, ensure that CoMeT is not currently running, then type

```
load CoMeTPrefs
prefs.FontSize = 8;
...
save CoMeTPrefs prefs
```

Naturally it is possible to make backup copies of the preferences file, and later switch between different preference schemes. To restore preference to factory defaults,

chose "Restore to default" from the preferences menu.

Saving `prefs` via the preference menu displays all preference fields at the command line. These include color schemes, the type of marker used for different types of nodes, the last beam/load/centroid properties entered, the snap grid spacing, several radii that determine how close one must click to a node to select it, layout spacing (such as the size of the thumbnail or padding between buttons), the last file used, the logical `LoadLastFile` flag that determines if the last file is automatically loaded during startup, and countless others.

Chapter 5

Quantification of Program Performance

As discussed in Chapter 1, CoMeT is tested in three main areas. First, elementary beam responses are verified via comparison to CosmosWorks® in Sec. 5.1. Next, a complex system, the Hexflex™ positioning flexure, is compared analytically and experimentally in Sec. 5.2. Finally, a user interaction study is presented in Sec. 5.3.

5.1 Element Model Verification

5.1.1 Rectangular Beams

The beam used to verify the correctness of CoMeT rectangular beam calculations, shown in Fig. 5-1, had a length of 6in, a width of 1in, and a height (depth) of 2 inches. The material properties were chosen as those of 6061-T6 aluminum, with a Young's Modulus of 10,000 KSI and a Shear Modulus of 3,770 KSI. Yield strength and density affect neither stress nor displacement calculations in this section.

Table 5.1 summarizes the variation in predicted beam displacement between

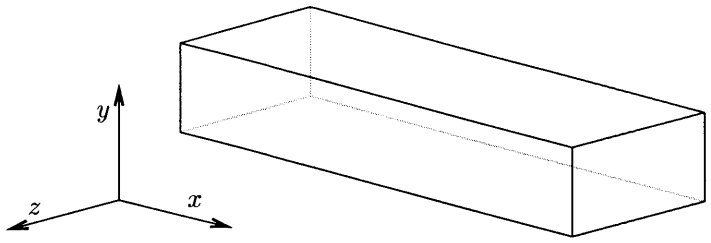


Figure 5-1: Rectangular beam used for FEA verification

Table 5.1: **Rectangular beam compliance discrepancies**

%	F_x	F_y	F_z	M_x	M_y	M_z
x	0.9					
y		2.4				1.6
z			0.7		1.4	
θ_x				10.0		
θ_y			-0.4		8.6	
θ_z		1.5				5.5

CoMeT and CosmosWorks. The blank fields correspond to force-displacement pairs that are analytically de-coupled. Cosmos consistently predicted these motions to be four to six orders of magnitude less than the principal motions, but never zero due to roundoff. Reporting an error of 100% would have been entirely misleading, hence no values were recorded.

Most results agree to better than 3%. In fact, with the exception of θ_y rotation under y-forcing, CoMeT always predicts slightly more motion than CosmosWorks, which is consistent with the theory that finite degrees of freedom will always put additional, artificial constraints on a structure's deformation response, resulting in less compliance than the continuum limit allows. Furthermore, Maxwell's Theorem of Reciprocal Forces dictates that the stiffness matrix of any element be exactly symmetric. Since CoMeT is precisely SPD, the asymmetry in deviation percentage is solely a consequence of CosmosWorks. It therefore appears that the larger part of the errors in Table 5.1 arise outside of CoMeT. In any case, a 3% discrepancy is

excellent agreement for engineering analysis.

The three moment rotation responses warrant further explanation. The version of CosmosWorks employed in this study allowed neither the application of pure moments nor the direct measurement of rotations. Instead, equivalent force couples had to be applied, and rotations derived from the fundamental relationship

$$\delta = \theta \times \mathbf{r}. \tag{5.1}$$

By probing displacements at two points and dividing their difference by their separation, rotation at the beam ends could be obtained. Euler beam theory assumes that cross sections remain planar, hence any two distinct probings on a given surface should produce the same angular displacement. However, this was not exactly true for the CosmosWorks simulation. In some cases, fringes appeared on the beam edge where the force couple forces were applied. The remaining elements remained planar, and rotation was derived from their motion. Perhaps a closer correlation could be achieved if displacements and rotations were averaged over the entire surface.

Next it should be noted that the rectangular beam in Fig. 5-1 has an aspect ratio of $3:1$ for non-planar loads. St. Venant's principle states that localized irregularities, such as those introduced by a beam's specific mounting constraints, average out three to five characteristic dimensions away from the disturbance. CoMeT loses in accuracy as beams become stubby and plate-like.

Finally, the CoMeT torsional model taken from Young and Budynas [17] is based on the assumption that cross sections only warp in the xy and xz planes. The CosmosWorks simulation clearly shows warpage at the beams's final yz section, since the beam end is in a state of plane stress. Torsional warpage is another localized effect that diminishes with increasing beam length. Furthermore, a round cross section will not warp, and the more square a beam section, the less error remains to be expected. In summary, rectangular beam elements match FEA models within three percent, but caution is advised wherever very short beams or torsional loads are present. In those cases, the actual geometry is likely to be sensitive to geometric details that CoMeT's

analysis does not capture.

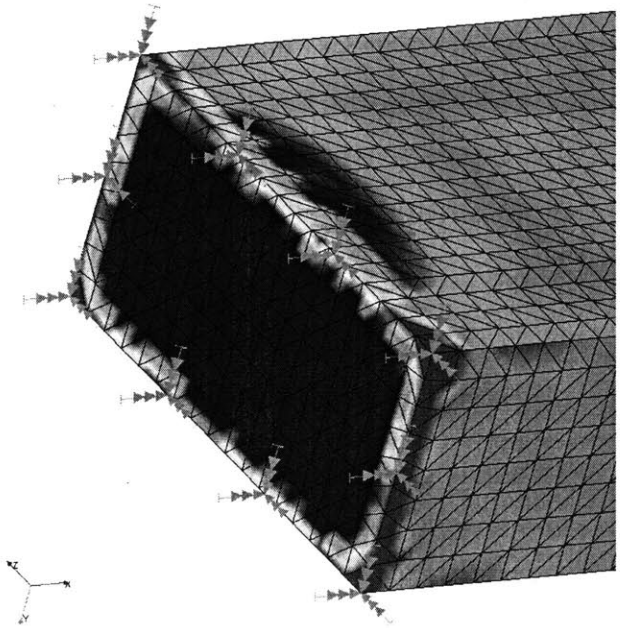


Figure 5-2: Typical localized stress fringe

Stress analysis was, generally speaking, more difficult than compliance analysis. All beam elements in CosmosWorks showed stress and displacement fringes to varying degrees, localized aberrations in the first and last section of elements. A typical stress fringe is show in Fig. 5-2. Since end displacements are the integral of the strains in a beam, fringes had relatively minor effects on displacements, as disturbances were averaged out over the structure. However, the maximum stress in a beam is a local quantity, sensitive to local conditions. In many cases, there were extremely localized stress peaks that determined the overall maximum stress. Although some of these peaks may reflect actual stress concentrations, many are clearly non-physical near-singularities.

In order to convey both the overall stress pattern and the localized stress peaks, Table 5.2 includes two blocks for comparison. The first, "σ_{max}, interpolated," shows data interpolated (or extrapolated) from the general stress field to avoid stress fringes. The second, titled "σ_{max}, actual," simply includes the highest stress that occurred

Table 5.2: **Rectangular beam maximum stress discrepancies**

	σ_{max}, CoMeT	σ_{max}, interpolated		σ_{max}, actual	
	PSI	PSI	%error	PSI	%error
F_x	0.50	0.50	0.0	1.06	-52.8
F_y	18.00	17.83	0.9	23.67	-24.0
F_z	9.00	8.95	0.5	14.91	-39.6
M_x	3.53	3.52	0.2	67.63	-94.8
M_y	1.50	1.50	0.0	63.25	-97.6
M_z	3.00	3.00	0.0	35.50	-91.5

anywhere in the CosmosWorks structure.

Elementary continuum mechanics teaches that maximum stresses at the outermost fibers of a beam remain constant under pure bending, which, with the exception of the aforementioned fringes, was indeed the case in CosmosWorks. Under force loading, moments and corresponding stresses should vary linearly along the length of a prismatic beam. Again, this was seen to be true for all but the end cross sections in Cosmos. Therefore, relatively high confidence may be placed in the "interpolated" columns, since many data points confirm their values. The absolute stress maxima vary for all the previously discussed reasons, but are included for the sake of illustration. Clearly the force couples (used to represent moments) led to unlikely stress spikes, usually at the corners of the beam.

Should the last two columns of Table 5.2 then be dismissed? The answer is clearly no. In point, Cosmos predicts a stress concentration factor of 2.12 under tensile loading, whereas rigorous stress analysis shows that the combined stress state of the corner element would exactly double the von Mises stress. The element in CosmosWorks was fixed rigidly at its left end, which led to xy and xz shear at the edges. If the rectangular beam element were indeed rigidly mounted to a much stiffer material, stresses would locally double. However, it could be argued that a monolithic compliant mechanism would never allow such hard constraints, and the interpolated stresses are more realistic. Table 5.2 therefore gives us a sense of the structure's

stress sensitivity to end mounting conditions. Only intuition and experience can distinguish an actual stress raiser from an FEA singularity. CoMeT avoids the mathematical pitfalls of finite element modelling, but fails to capture physical stress concentrations. Nevertheless, it should accurately predict relative stress levels in competing designs, and stress concentrations can usually be minimized in the detailed design stage through common design practices. Since precision engineers are typically more concerned about motions than stresses, they may take CoMeT deformation predictions for what they are, but apply standard stress safety factors to the final design.

5.1.2 Round Beams

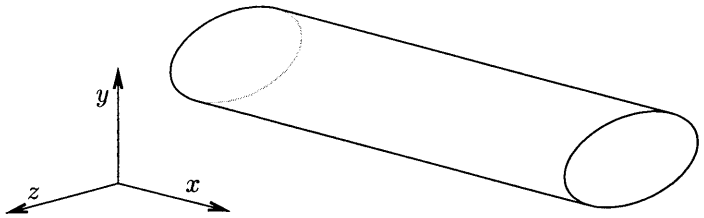


Figure 5-3: Round beam used for FEA verification

The CoMeT round beam equations generally do not differ much from their rectangular equivalents, a fact reflected in similar stress and displacement accuracy results. The round beam under consideration in this subsection, pictured in Fig. 5-3, had the same height, width, and length as the beam in Fig. 5-1. This allows for direct comparisons between the two.

Table 5.3 summarizes the round beam displacement prediction variations analogously to Table 5.1. Results generally appear to be better than in the rectangular section, which has several definite causes. First and foremost, a round beam is intrinsically smooth, which reduces computational difficulties. Secondly, force couples for moment simulations were applied indirectly, by means of a small rectangular block at the beam's end. This block "absorbed" many of the fringes, leaving the beam in

Table 5.3: **Round beam compliance discrepancies**

%	F_x	F_y	F_z	M_x	M_y	M_z
x	0.8					
y		2.1				1.6
z			1.0		1.2	
θ_x				1.7		
θ_y			1.2		-1.2	
θ_z		1.9				1.4

a purer state of loading.

The most notable relative improvement is in torsional compliance. Although the beam is twice as high as it is wide, its surface is sufficiently convex, and warpage is minimal. Again, a beam of circular cross section theoretically does not warp at all, assuming proper boundary conditions.

As before, almost all deviations are positive, as CoMeT predicts more displacement than the commercial FEA package. The negative rotational discrepancy due to non-planar bending deviates from this pattern slightly, but not significantly. In any case, round beam deformations agree with CosmosWorks to better than three percent under all loading scenarios.

Table 5.4: **Round beam maximum stress discrepancies**

	σ_{max}, CoMeT	σ_{max}, interpolated		σ_{max}, actual	
	PSI	PSI	%error	PSI	%error
F_x	0.64	0.64	-0.1	0.93	-31.4
F_y	30.56	32.30	-5.4	30.98	-1.4
F_z	15.28	15.29	-0.1	16.60	-8.0
M_x	4.41	4.40	0.4	6.07	-27.3
M_y	2.55	2.55	-0.1	5.81	-56.2
M_z	5.09	5.11	-0.2	6.69	-23.9

Round beam stress predictions offer an even better improvement over its

rectangular equivalents, due to the absence of sharp edges. All interpolated values are in excellent agreement with conventional theory. Several rows in Table 5.4 even show very little discrepancy between the interpolated and actual maximum values. The largest discrepancies can be attributed to local states of plane stress and minor fringes at the beam boundaries.

5.1.3 Tapered Beams

Unlike all the other beam elements, which made use of age-old established formulas, tapered beam deflections are somewhat of an approximation. When the moment curvature relation expressed in Eqns. (2.66) and (2.67) is integrated, an implicit assumption is that the limit of a series of short rectangular beam segments with the appropriate widths and heights converges to the tapered beam response. This cannot be exactly true, since each cross section presumably has a linearly varying vertical stress distribution. If this were the case, the beam surface would be in disequilibrium, even in the limit of infinitely many sections. There has to be a shear force to compensate the beam widening, a force that is entirely absent in prismatic beams, and that leads to complex warping. However, the irregularities near the surface should remain minor compared to the tensional stress effects throughout the rest of the beam volume, as long as the surface varies gradually [17, p. 158].

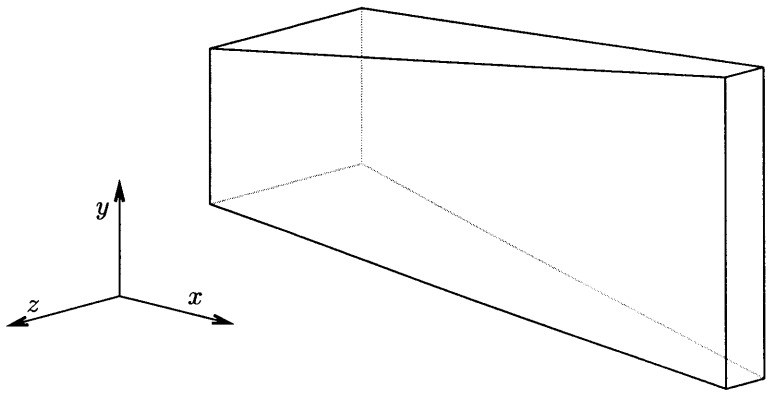


Figure 5-4: Strongly tapered beam that exhibits many plate-like qualities

Two tapered beams were tested to illustrate the range of validity of the CoMeT model. Fig. 5-4 shows a "stout" tapered beam whose final width is only two thirds of its length.

Table 5.5: **Geometric parameters of the studied tapered beams**

	Stout		**Slender**	
length	6 in		6 in	
	width	height	width	height
initial	2 in	2 in	$\frac{1}{2}$ in	1 in
final	4 in	1 in	1 in	$\frac{1}{4}$ in
slope	9.5°	4.8°	2.4°	3.6°
attenuation	2×	2×	2×	4×

The tapered beam geometric parameters are listed in Table 5.5. Note that "slope" refers to surface slope; the wedge angle is twice the surface slope. Although 9.5° and 4.8° do not appear to be excessive angles, combined with a final beam length to width ratio of 1.5, they suffice to make the stout beam displacements and stresses quite inaccurate.

Table 5.6: **Stout tapered beam compliance discrepancies**

%	F_x	F_y	F_z	M_x	M_y	M_z
x	-8.5					
y		-44.7				-42.4
z			-12.9		-11.5	
θ_x				22.5		
θ_y			-9.4		1.5	
θ_z		-44.4				-27.3

Table 5.6 shows the stout beam analysis results. Non-planar compliances (where height is critical) are within 15% of CosmosWorks calculations, which is considerably better than planar compliances, which vary by as much as 45%. Torsion caused

massive warpage in all directions, and a large block of material had to be added to the end of the beam to even ensure a level plane whose angle of rotation could be measured. It also warrants mentioning that CoMeT compliance predictions fall *below* the FEA value. The stout tapered beam really resembles a plate, and CoMeT artificially constrains it to behave like a beam.

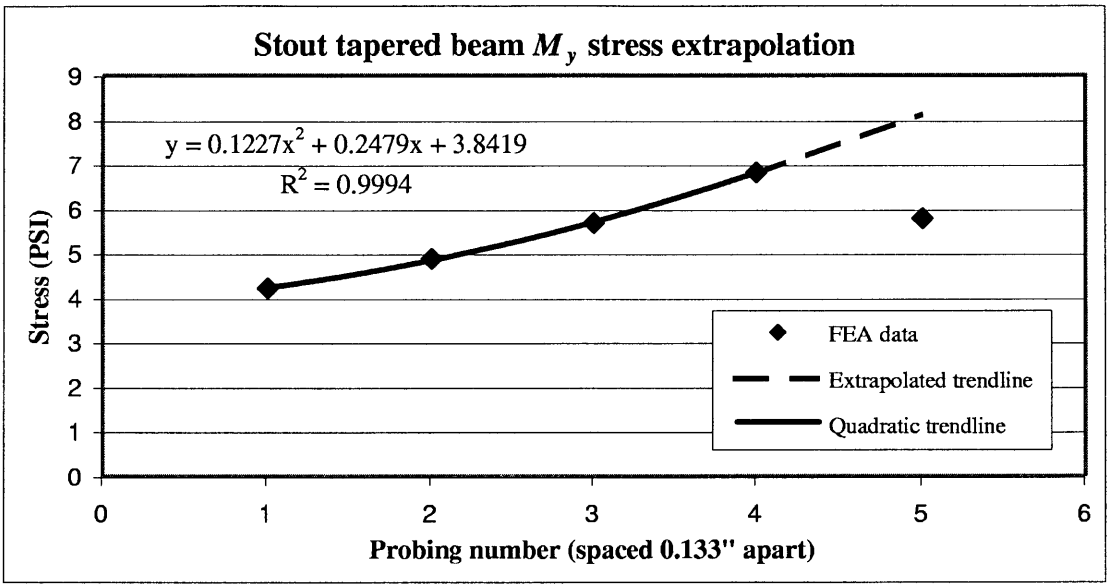


Figure 5-5: Typical stress extrapolation curve

In rectangular and round beams, stress fringes were circumvented via linear extrapolation. Curved beams, unlike their prismatic counterparts, feature nonlinear stress distributions. Figure 5-5 shows a typical extrapolation curve (for the stout tapered beam, y-deflection under M_y-loading). Displacements were probed at regular spatial intervals, dictated by the mesh grid, approaching the beam boundary. Probing number five is exactly on the boundary, and clearly takes an unusual dip, the result of a stress fringe. Although probing four would not result in much error, a procedure that does better justice to the CoMeT analysis is a local quadratic extrapolation. A parabola is fit to points one through four and extended to point five. The same procedure was repeated for each loading case, resulting in Table 5.7, as well as Tables 5.9 and 5.13.

Table 5.7: **Stout tapered beam maximum stress discrepancies**

	σ_{max}, CoMeT	σ_{max}, interpolated		σ_{max}, actual	
	PSI	PSI	%error	PSI	%error
F_x	0.67	0.69	-3.7	0.77	-14.0
F_y	4.50	8.58	-47.6	10.93	-58.8
F_z	4.61	5.71	-19.2	8.78	-47.5
M_x	5.48	5.20	5.4	7.31	-25.0
M_y	8.00	8.15	-1.8	8.65	-7.5
M_z	1.33	1.30	2.6	3.26	-59.1

Stress (Table 5.7) also indicates that the Euler beam assumptions do not fully apply to the stout tapered beam. However, several stress levels are much better than one might expect given the corresponding displacement inaccuracies. The reason appears to be that cantilever loading induces radically different stress fields, whereas bending, tension, and torsion do not. Or, at a minimum, since the highest stress levels always occurred at one of the beam ends, the latter loading cases do not change the maximum boundary stress one would find in a rectangular beam with the same cross section. Nevertheless, a natural system will have mixed loading, and a reliable stress level prediction would require detailed FEA studies involving solid elements.

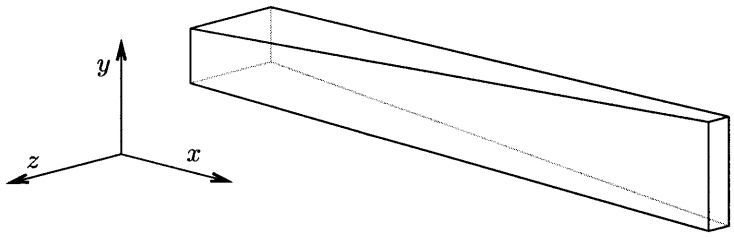


Figure 5-6: Slender tapered beam to which CoMeT's theory applies

Figure 5-6 shows a slender tapered beam that meets the assumptions imposed by CoMeT more closely. Its geometry is also given in Table 5.5. Although its attenuation factor exceeds that of the stout beam, its minimum length to width ratio is now 6, which minimizes end effects that are usually conspicuous in short beams.

Furthermore, its surface slope is less than half of what it was before.

Table 5.8: **Slender tapered beam compliance discrepancies**

%	F_x	F_y	F_z	M_x	M_y	M_z
x	0.2					
y		0.7				0.6
z			0.4		0.4	
θ_x				11.3		
θ_y			-0.1		2.8	
θ_z		0.1				0.0

As advertised, the slender beam compliance, shown in Table 5.8, is in much better agreement with CosmosWorks' calculations. In fact, agreement even exceeds that of the rectangular beam in Table 5.1. This is likely a consequence of the larger length to width ratio, and an equally lank rectangular beam should exhibit even better correlation. The 2.8% M_y-θ_y discrepancy is due to minor fringing across the $\frac{1}{4}$" wide section, which was too small to allow proper averaging. The torsional discrepancy is more serious, but for a justifiable reason. While the other beam models rely on an assumed σ_x stress field, torsion produces a radial (τ_{rt}) shear field. The beam taper introduces shear stresses not captured in the model, and these interact strongly with the torsional field.

Table 5.9: **Slender tapered beam maximum stress discrepancies**

	σ_{max}, CoMeT	σ_{max}, interpolated		σ_{max}, actual	
	PSI	PSI	%error	PSI	%error
F_x	4.00	4.18	0.0	4.00	-4.2
F_y	140.36	159.23	2.6	144.00	-9.6
F_z	72.84	90.81	-1.1	72.00	-20.7
M_x	66.91	66.91	4.0	69.55	4.0
M_y	81.40	97.18	17.9	95.99	-1.2
M_z	24.58	30.93	-2.4	24.00	-22.4

Slender tapered beam stress discrepancies are shown in Table 5.9. Errors are mostly below 3%. The M_y-θ_y stress discrepancy of 17.9% is again a result of fringing along the thin $\frac{1}{4}$" dimension. Torsion exhibits its usual problems by raising stresses by 4%, which is not disconcerting.

If stout tapered beams vary so substantially from the CosmosWorks simulations, could there be errors in the CoMeT equations themselves? To test this hypothesis, the stout tapered beam pictured in Fig. 5-4 was simulated in CoMeT using 100 short rectangular beams in series, whose width and height matched the the average cross sectional properties at their respective locations. Although each of these beams had a deplorable length to width ratio, the idealized boundary conditions should ensure that this error does not amplify beyond what a single tapered beam would experience.

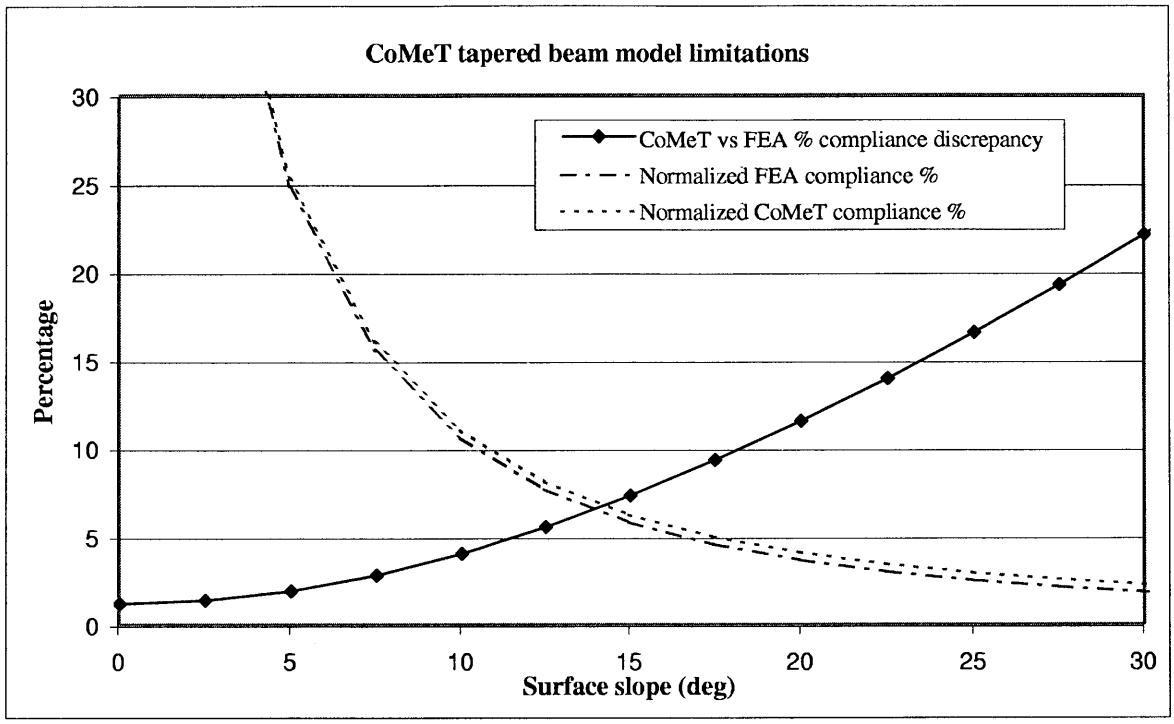
Table 5.10: **Tapered beam displacement compared to 100 discretized rectangular beams**

%	F_x	F_y	F_z	M_x	M_y	M_z
x	0.00					
y		0.00				0.00
z			-0.01		-0.01	
θ_x				0.02		
θ_y			-0.01		0.02	
θ_z		0.00				0.00

Table 5.10 shows the results of the discretized beam study. Clearly the discrepancies are negligible. In fact, they are on the order of $(1\%)^2$, which is to be expected with 100 segments that match the tapered beam to first order. Further trailing digits have been suppressed because they are meaningless in this context. Table 5.10 shows that, although the underlying tapered beam modelling assumptions may break down for stout tapered beams, the equations are mathematically correct.

The question remains, if slender tapered beams agree to theory, and stout beams vary somewhat, to which beams can CoMeT be applied with confidence? Unfortunately there is no easy answer, since there are so many parameters to vary,

Table 5.11: **Tapered beam displacement validity as a function of surface slope**



and so many compliance responses to check. Generally speaking, applicability of taper approximation is a function of the surface slope. Naturally a strong surface slope implies a low length to maximum width ratio, which causes problems in itself. Table 5.11 shows the results of a series of FEA studies in which initial width was systematically varied. Length was 6", height was uniform at 1" (height drops out of the equation), and final width was also 1". Initial width was solved so as to vary the surface slope in 5° increments. Discrepancy is only computed for y-compliance to due to F_y-forcing. In other words, the load was applied in the plane of the taper.

Table 5.11 suggests that a beam with a 12° taper (24° wedge angle) differs by 5%, whereas 18 degrees represent the 10% threshold. This may be true for the case at hand, but a number of factors can increase the discrepancy. Obviously the doubly-tapered beam in Table 5.6 shows much worse correlation, and no simple rules are apparent for double taper interaction. Usually a plate-like response can

be anticipated via inspection. If a strongly tapered beam plays a critical role in a compliant mechanism, a formal FEA study should be performed on the final design.

If demands are not so stringent, the picture looks considerably less bleak. Table 5.11 also shows two normalized compliance curves. The absolute compliance values predicted by CoMeT and CosmosWorks ("FEA") were normalized by the Cosmos compliance for a 0° slope beam. This was intended to put the absolute compliance "into perspective" with the rest of a typical mechanism. If another rectangular beam of equal length was attached to the thin end of the tapered beam, the dot-dashed line roughly indicates the percentage of total compliance affected by the tapered beam. Even though the relative separation between the Cosmos and the CoMeT line grows, the absolute difference remains below one (normalized) percent. In other words, the tapered beams that are most in error are also the least compliant, and are unlikely to strongly affect the overall system response.

5.1.4 Curved Beams

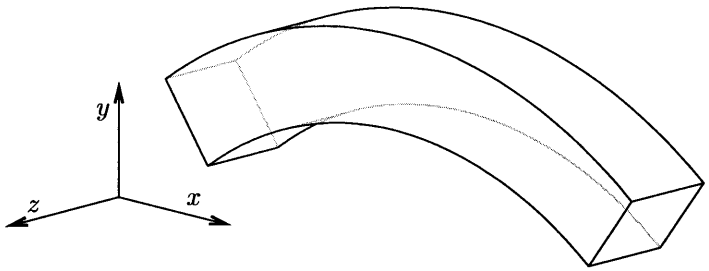


Figure 5-7: Curved beam used for FEA verification

The curved beam used to verify the equations in Sec 2.4.4 had a one inch square cross section and a nodal separation of six inches. The mean radius is five inches, giving an included angle ϕ of 73.74°.

The displacement error chart shown in Table 5.12 shows excellent correlation. Again, the matrix should be symmetric about its main diagonal. The F_z–θ_y deviation of -2.3% therefore stands in conflict with the M_y– z deviation of 1.1%, pointing

Table 5.12: **Curved beam compliance discrepancies**

%	F_x	F_y	F_z	M_x	M_y	M_z
x	1.6	0.8				0.8
y	1.1	1.4				0.8
z			0.0	1.6	1.1	
θ_x			1.1	0.0		
θ_y			-2.3		2.4	
θ_z	0.9	0.6				0.0

to problems in CosmosWorks. M_x–correlation, usually a weak point of CoMeT, stands at less than 0.05%. This is partially because M_x induces very little actual torsion; most of the deflection is due to bending in the first and last third of the beam. However, any correlation beyond half a percent appears more fortuitous than consistently achievable.

Table 5.13: **Curved beam maximum stress discrepancies**

	σ_{max}, CoMeT	σ_{max}, interpolated		σ_{max}, actual	
	PSI	PSI	%error	PSI	%error
F_x	7.43	7.50	-1.0	7.51	-1.1
F_y	39.16	39.03	0.3	44.79	-12.6
F_z	35.77	37.43	-4.4	48.84	-26.8
M_x	5.93	8.69	-31.8	72.61	-91.8
M_y	6.00	7.60	-21.1	64.11	-90.6
M_z	6.43	6.45	-0.4	68.85	-90.7

The curved beam stresses were extrapolated in the manner explained for Fig. 5-5. Stresses under planar loading are clearly in excellent agreement, although the usual fringing occurs. As discussed in Subsection 2.5.3, non-planar stress fields are quite complex, and the neglect of Poisson effects leads to significant errors. Incidentally, the non-planar stress prediction would only degrade if a curved beam were replaced by many rectangular ones, since rectangular beams are not affected by Poisson

contraction. The higher the width to radius ratio, the more significant Poisson effects become. Fortunately, curved beam displacements are calculated independently of stresses, so deformations and stresses in adjacent beams would remain accurate.

5.2 Complex System Case Study: The Hexflex^{TM}

The Hexflex^{TM} six axis flexural positioning stage that motivated this thesis was built and tested by Mr. Gordon Anderson [5]. The CoMeT model, CosmosWorks model, and experimental data are compared in this section. The largest discrepancy between CoMeT and CosmosWorks was less than 4%, far below expectations. CoMeT and reality differed by no more than 9%, more than acceptable for a first-order design pass. In the process of testing, valuable insights were gained into flexure design and CoMeT limitations.

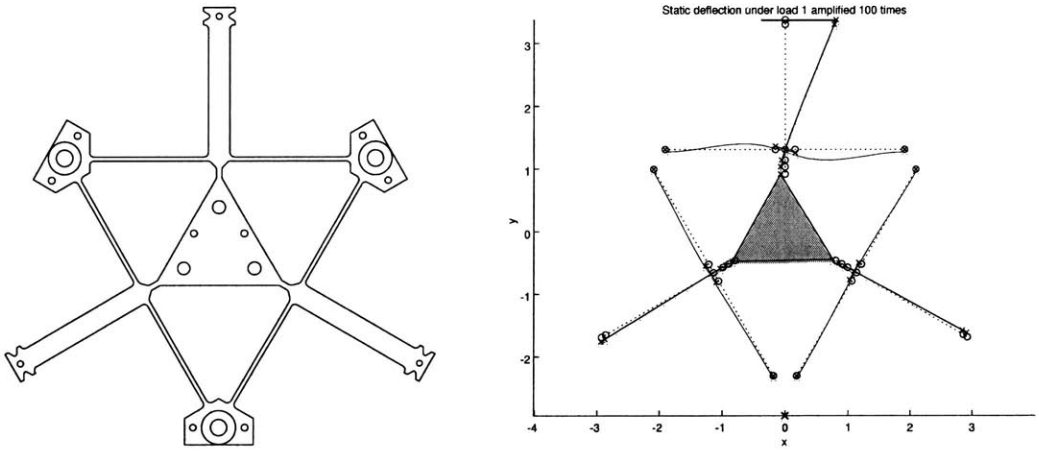


Figure 5-8: The piezoelectrically actuated Hexflex^{TM}. Left: outline of CAD model used for CosmosWorks and manufacturing. Right: CoMeT model with superimposed planar actuation deformation response

Fig. 5-8 Shows the version of the Hexflex^{TM} optimized for and used in piezoelectric actuation experiments. The CoMeT model clearly simplifies many features, including rounds, holes, and fillets. The central triangle is represented as a rigid plate with short tapered beams connecting to the necks. Eighteen beams are in the model, not

counting the short breaks in the tabs just before the actuation point, which were used to capture actuator misalignment in the sensitivity study. The model parameters were adjusted to "as manufactured" values, but symmetry was retained.

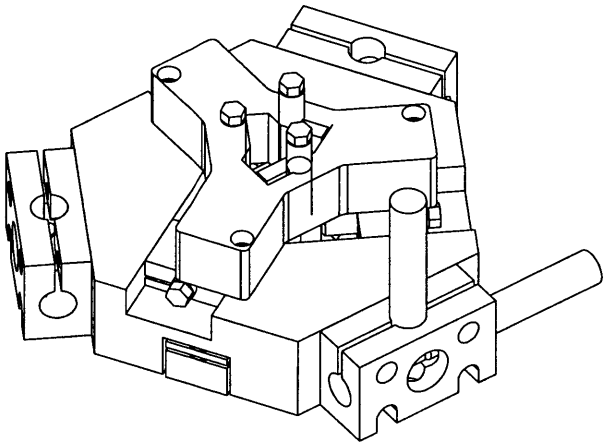


Figure 5-9: Piexoflex experiment setup

Fig. 5-9 shows the experimental Hexflex^{TM} setup. The two long cylindrical devices are planar and non-planar piezoelectric actuators. The Hexflex^{TM} tab they push on is not visible. Six capacitance probes were used to obtain accurate position feedback, five of which are in view and identifiable by their hexagonal ends. For experiment details, please consult Mr. Anderson's thesis [5].

Planar actuation results are summarized in Table 5.14. The actuators are numbered 1 through 3 viewed clockwise from above, starting at the upper actuator. They are oriented as shown in Fig. 5-9, patterned radially. Subscripts p and z denote planar and non non-planar actuators, respectively. The first three sets of tests, titled "A1$_p$" through "A3$_p$", are radially symmetric and should produce identical θ_z motions. In the "A23$_p$" test, actuators 2 and 3 were moved equal amounts in opposite direction, inducing pure y displacement. The final test, "A123$_p$", spun the centroid about its center.

Comparing CoMeT to Cosmos, we find deviations of 1.5% or less. The result is surprising given the list of possible sources of inaccuracy — neglected rounds, strongly tapered beams (30° and 60° surface slope), and short beams in the critical

Table 5.14: **Planar Hexflex^{TM} actuation correlation**

%		$A1_p$	$A2_p$	$A3_p$	$A23_p$	$A123_p$
x	Linearity error	0.3	2.1	1.2		
	CoMeT vs Cosmos	1.3	1.2	1.2		
	CoMeT vs actual	3.0	0.0	4.0		
	Cosmos vs actual	1.7	-1.2	2.8		
y	Linearity error		0.7	0.5	0.4	
	CoMeT vs Cosmos		1.3	1.3	1.3	
	CoMeT vs actual		2.6	3.9	2.4	
	Cosmos vs actual		1.3	2.6	1.1	
θ_z	Linearity error	0.9	0.8	2.1		0.3
	CoMeT vs Cosmos	-1.5	-1.5	-1.5		0.2
	CoMeT vs actual	2.1	-1.3	-2.4		2.8
	Cosmos vs actual	3.6	0.2	-1.0		2.6

neck region — and single element inaccuracies (Table 5.1) that already exceed 2%. It appears that the taper stabilizes the neck boundary conditions in Cosmos, while remaining so much stiffer than the neck that CoMeT's tapered beam approximation breakdown has few adverse effects.

Reality introduced additional discrepancies. A linearity error of 2% should be an indication of the order of magnitude of experimental uncertainty. The 4.5% difference between supposedly symmetric rotations under $A1_p$ and $A3_p$ actuation is another. In any case, the largest planar CoMeT-experiment discrepancy was 4.0%, statistically no better than Cosmos' model, which differed by up to 3.6%.

Non-planar actuation was slightly more problematic. The three tabs were individually actuated in the vertical direction for the middle set of columns in Table 5.15, then in unison, producing pure z-translation results in the final column.

The maximum difference between CoMeT and Cosmos now rose to 4% for single-tab actuation vertical displacement, although rotations matched an astounding 0.1%. The rotation differences are really only a single data point, since all tabs were perfectly symmetric in the analytic models, and θ_x rotation is no different from θ_y when

Table 5.15: **Non-planar Hexflex^{TM} actuation correlation**

%		$A1_z$	$A2_z$	$A3_z$	$A123_z$
z	Linearity error	1.0	1.5	1.3	1.0
	CoMeT vs Cosmos	-3.9	-3.9	-3.9	-3.5
	CoMeT vs actual	8.3	-8.8	5.5	3.0
	Cosmos vs actual	12.7	-5.1	9.8	6.7
θ_x	Linearity error	0.5	0.5	0.8	
	CoMeT vs Cosmos	0.1	0.1	0.1	
	CoMeT vs actual	3.3	-0.9	5.6	
	Cosmos vs actual	3.2	-1.0	5.4	
θ_y	Linearity error		0.6	0.6	
	CoMeT vs Cosmos		0.1	0.1	
	CoMeT vs actual		-0.4	3.7	
	Cosmos vs actual		-0.5	3.6	

the coordinate frame is rotated accordingly. The relatively inferior z-performance is not unexpected, since the necks have a smaller length-to-height than length-to-width ratio, torsion is now present, and tapered beams are less stiff when forced perpendicular to the plane of taper.

Non-planar experiments were generally less successful than the previous set. Over 17% scatter in theoretically symmetric single-tab actuation is indicative of hardware problems. Although manifested physically, the root cause may be explained on theoretical grounds. First, the non-planar geometric deamplification factor was 44, three times stronger than the planar factor of 15. Parasitic motions of the same absolute magnitude would therefore appear three times worse. Furthermore, the nonplanar deformation mode is much more sensitive to both errors in neck geometry and compliance in the triangular stage. Finally, the waterjet cutting process used to manufacture the flexure is most likely to introduce defects in the projected shape, which lies in the xy-plane. These would, in turn, most affect non-planar motions.

Given the experimental uncertainty, a maximum CoMeT deviation of 9% is actually quite good. In fact, CosmosWorks is actually off by quite a bit more,

namely up to 12.7%. It would be presumptuous to claim that CoMeT actually beats traditional FEA, but it should be fair to say that the two finite element varieties are equally apt in predicting actual motions. A designer can safely approximate geometry in CoMeT, expecting agreement to lie within 10% of actual field data, so long as the model matches the actual system.

5.3 User Interaction Study

CoMeT's computational speed is unquestionably better than that of mesh-based finite element programs, but computation time is seldom the dominant factor in compliant mechanism development. The real benefits of using CoMeT are distributed over the entire design cycle, as development speed is increased for a number of reasons. Representing compliant mechanisms in CoMeT format forces one to confront the essence of a design concept, as distracting details are eliminated. Tendencies are more easily identified after second-order effects are "filtered" out. While traditional FEA tend to calculate only elementary quantities like displacements, stresses, and strains, CoMeT computes high-level variables that stimulate high-level thought. These normalized quantities and dimensionless numbers ($\mathbf{K_A}$, $\mathbf{S_x}$, $\mathbf{F_{A,\,max}}$, η, T, and sensitivity, for example) absorb simple scaling relationships, exposing the intrinsic characteristics of a mechanism's topology or geometry. Demands on the user's attention span are reduced, leaving more mental "processing" capability for system-level considerations. Most importantly, CoMeT minimizes much of the frustration and monotonous work associated with traditional analysis, kindling a designer's latent curiosity and passion for understanding.

The human interaction study has revealed physical indications that may not prove, but definitely strengthen the claims made in the previous paragraph. While the numerical results are informative and less open to interpretation, the subjective observations are all the more revealing.

In the study, a human subject solved four similiar compliant mechanism optimiza-

tion problems, using CoMeT and CosmosWorks alternatingly as supporting software. The subject had less than ten hours of experience in CosmosWorks and only cursory prior exposure to CoMeT. The first two problems were presented in a low-pressure context, giving the subject an opportunity to gain familiarity with the software, problem type, and wording. After a brief intermission, the second set of problems were administered, this time under strict timing and administrator passivity.

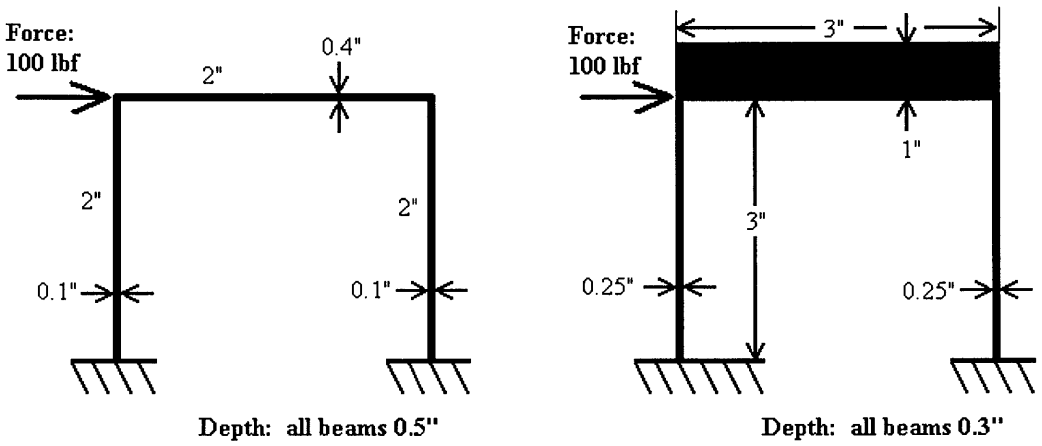


Figure 5-10: Sketches of compliant mechanisms to be optimized by CoMeT (left) and CosmosWorks (right) in the second round of user interaction study problems.

Each problem involved creating, analyzing, and optimizing a planar three-beam compliant mechanism. The optimization task demanded that two objectives be met through the action of two design parameters. The problem was partially coupled, so that constraints had to be satisfied in a specific order. Fig. 5-10 shows the topology used in the second round of problems. The full problem statement handouts are included in Appendix A.

Table 5.16 summarizes the results for each task and test. The subject was able to finish his task approximately twice as fast using CoMeT in both series. The second CoMeT optimization task time was inflated by an unfortunate coincidence to be covered later. A chronological discussion of the proceedings follows.

After an intensive thirty minute CoMeT tutorial and thirty more minutes of tackling the problem, the user solved his first optimization problem successfully.

Table 5.16: **User Interaction Times by Task**

Time (min)	Practice		Second Set	
Program	CoMeT	Cosmos	CoMeT	Cosmos
Draw	9	11	4:10	8:30
Analyze	3	19	1:35	3:25
Optimize	18	38[†]	18:20[‡]	31:30
Total	30	68	24:05	43:25

[†] With administator assistance

[‡] Delayed by hand calculation error

The mechanism was a Y-shaped flexure to be optimized for rotational performance, and somewhat more difficult to draw and optimize than the table-shaped device shown in Fig. 5-10. Use of the optimization tool was explicitly forbidden. Although the administrator made no formal attempts to be impartial, his contributions were insubstantial.

The second problem used the same topology as the first, with only parameters changing values. The analysis task demanded the same three quantities as the CoMeT version, but had to be derived from other measurable quantities this time. The hand calculations and the process of learning new probing techniques, despite administrator assistance, took a staggering 19 minutes.

The optimization task elicited the question "Do I have to do this?" from the subject. Assured of the importance of equal exposure to both programs, he proceeded, but struggled. Two impediments to progress were identified. Although the solution and initial geometry were both dominated by the customary plane strain, an unfortunate guess put the mechanism in the plane stress regime. The effects, exacerbated by overall volatility,[1] temporarily reversed the otherwise monotonic trend, leading subsequent guesses further astray. The second impediment was

[1]The subject had been instructed to always use the default mesh size, since preparatory explorations had revealed that maximum stress predictions for identical structures varied by 25% over the range of allowable grid sizing. Compliances varied by 5% over the range. It is likely that part of this volatility remained.

repeated loss of concentration. Evaluating a guess took so long that the subject lost his train of thought.

After thirty gruelling minutes, the administrator was forced to reveal one parameter, and advise how the second could be found using linear scaling. Two iterations later, the solution was found, for an overall optimization time of 38 minutes.

Following a short lunch break, the second batch of tests (included in Appendix A) were administered. The CoMeT drawing task was delayed about one minute by attempts to create a load in the default "modify load" mode. The subject was preoccupied by the surmise of a bug in CoMeT, since selecting a free node had no visible effect, and he had been advised of an ostensibly similar problem during the tutorial.[2] The entrenched habit of drawing before dimensioning, as is common in CAD programs, marginally increased the drawing task time some more.

The analysis section was completed in 95 seconds, and prompted a brief off-the-record discussion. The subject inquired why the slight vertical displacement of the actuated node was positive, when his intuition indicated otherwise. The effect would have been unnoticable in traditional FEA, being less than 0.1% of horizontal motion, but so prompted an interesting discussion of the moment distributions in each beam.[3]

The optimization task time was marred by an unfortunate mistake. Although some confusion surrounds the exact nature and timing of the incident, there is no doubt that it delayed the study considerably. The subject identified the first design parameter, then attempted to apply the proportional scaling technique learned during the previous Cosmos optimization. A fraction was inadvertently inverted, prompting a bad guess. The result, obtained 7:10 min into the optimization, induced further confusion. Since result and input were falsely deduced to vary inversely, the subject decided to apply a generalized linear interpolation. Shortly after the 15:20 min mark, the approach was abondonend in favor of trial and error. Three minutes later, the final solution was found.

---

[2] A problem with the load `HitTest` property had prevented selectability of the loaded node from its immediate vicinity.

[3] As beam force probings quickly revealed, the vertical beam closest to the load is in tension, since it has to help cancel the moment exerted by the load relative to ground.

The fourth and final exercise in CosmosWorks was started immediately afterward. The subject's creation time benefitted from use of the SolidWorks mirroring function, but suffered when a small inadvertent sketch entity precluded extrusion and could not be immediately found. The analyzation task was completed in a brisk 3:25min, showing that the subject had learned to derive the required quantity $\mathbf{x_{A,\,max}}$ efficiently. The second Cosmos optimization was completed, without special incidents, in 31:30min.

At face value, the CoMeT procedure took 45% less time than the CosmosWorks equivalent. Although it is obvious that the former test was unduly delayed by the irregularity, it is not sensible to speculate what results would have been in an incident-free setting. There are too many factors to consider, and the study imposed many debatable limitations to begin with. For example, the optimization tool would have been ideally suited for the problem at hand, but was excluded for the sake of simplicity.

Although of limited scope, the user interaction study confirmed that CoMeT consistently produces results faster than the leading commercial FEA program. Furthermore, anecdotal evidence suggests that CoMeT has substantially higher didactic value, encouraging substantive questions and quickly providing the answers. Field experience from the PSDAM lab can only add to the list of subjective benefits. Only time can reveal all of CoMeT's advantages, but judging on current evidence, there are plenty.

Chapter 6

Conclusions

6.1 Recommendations for Future Work

CoMeT can be further developed until it rivals any commercial package on the market. Its flexible, modular structure could easily handle additional linear analysis. On the other hand, its efficient solution process lends itself to iterative, nonlinear studies. A number of interface enhancements and advanced modules may also prove desirable.

Most first-order design considerations are of a linear nature, and fit seamlessly into CoMeT's scheme. As mentioned in Sec. 2.7, mixed force/displacement constraints could be solved using post analysis that builds on the existing block stiffness approach. Thermal expansion effects could be incorporated, although the distributed nature of thermal loads would require some thought. Different elements of a structure could be allowed to have different material properties (Young's modulus, shear modulus, yield stress, and density) by expanding these scalar quantities into vectors. Non-planar curved beam stress formulas could be made to include Poisson effects. Other linear analysis that would require more extensive work includes anisotropic material models, extensions to the library of elements, approximate nodal analysis, and dynamics.

Although initially intimidating, iterative nonlinear analysis may be essential to many innovative designs. In the future, some compliant mechanism concepts may

strive to derive first-order functionality from phenomena that are mathematically of higher order. For example, positioning flexures that hold their position via plastic set have been proposed. Design of such a system would most likely require plasticity, creep, geometric nonlinearity, and strain hardening analysis. Buckling may be used for bistable or multi-stable mechanisms such as switches and finite state positioning systems. Contact mechanics may produce variable stiffness, Coulomb friction may also be used to hold position, and the possibilities and analytic considerations of dynamics are endless. Traditional finite element packages may be adequate for iterative analysis if the nonlinear problems are decoupled from the remainder of the system, but most successful designs are simple, thus tightly integrated. CoMeT could be extended with the appropriate analysis modules as promising nonlinear mechanism concepts emerge.

The CoMeT editor was initially intended to function as a "sketch pad", taking approximate inputs. Recent work has shown that the bulk of analysis requires exact coordinate and property entry. It may therefore prove useful to create an extensive menu of drawing constraints. Alternatively, commercial sketching packages could be utilized to create files for the CoMeT core, if the appropriate converters were created. MasterCam®, for example, provides an excellent drawing interface that many engineers are familiar with.

The advanced modules could all be extended. A relatively simple, yet extremely useful addition would be the integration of MATLAB's `fgoalattain` command, found in the optimization toolbox, into the analyzer. This amazing function performs multivariate minimization subject to any number of linear or nonlinear equality or inequality constraints [3]. The constraints may be mixed and matched at will, so nearly any conceivable optimization is possible with minimal prior analysis.

The motion diagnosis tool performs a scalar η-T decomposition, which could be generalized into matrix form. If combined with energy methods and other considerations, it could then be used to generate topological predictions and recommendations. This was the initial idea behind motion diagnosis, but time constraints limited the work that has been done to date.

An analytic equation derivation module was once considered, but also became a victim of time constraints. MATLAB's symbolic math package could be used in conjunction with approximation techniques to generate analytic system models. These could be used as an alternative approach to numerical optimization. Analytic models often lead to more powerful insights than numerical ones. In a classroom environment, the analytic approximations could be used as a teaching tool. Designers could use the attained insights to design better and more advanced compliant mechanisms.

6.2 Proliferation

A journal article presenting highlights of the CoMeT program is scheduled for submission within the year. Mr. Anderson's experimental development of the Hexflex® has been submitted to the ASPE fall conference [10]. CoMeT will be advertised and distributed to professors in the MIT mechanical engineering curriculum. The software will either be licensed to Mathworks, Inc., or distributed freely on the world wide web.

6.3 Conclusions

This thesis documented the analytic development of CoMeT, a conceptual evaluation and detailed synthesis aid for the design of compliant mechanisms. Rectangular, round, tapered, and curved beams have been modelled successfully. Tests have shown that CoMeT calculations for for single element deflections are mostly within 3% of of those predicted by commercial finite element analysis. The discrepancies are a consequence of neglected shear and radial stresses, which are a second-order influence in flexural systems. Furthermore, an elaborate compliant mechanism, the Hexflex™ positioning system, was built and tested. The predicted centroid motion responses lie within 4% of traditional FEA and 9% of the experimentally measured

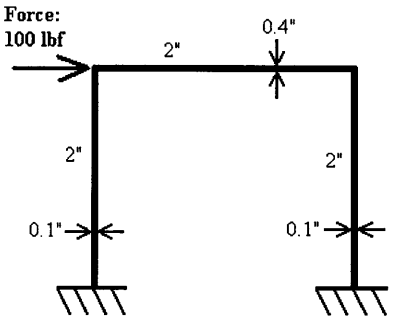
values, a more than adequate fit for a first-order design tool. Finally, a human subject with no relevant prior experience was asked to perform two multivariate flexure optimizations, once using CoMeT, the second on commercial FEA software. The CoMeT optimizations were performed in half the time, and proved to be more insightful. It has therefore been established that CoMeT is an accurate, user friendly, productivity-enhancing tool.

Appendix A

User Interaction Study Handouts

Compliant Mechanism Design Problem, CoMeT version

You are to create, analyze, and optimize the compliant mechanism below. Please read all instructions carefully. Good luck!



Depth: all beams 0.5"

Step 1: Create

Create the mechanism as shown. All beams are 2" long and 0.5" deep. The vertical beams are 0.1" wide, while the horizontal beam is 0.4" wide. A horizontal 100-pound force is applied to the upper left corner.

When you are done, check your work, notify your instructor, and proceed without further delay.

Step 2: Analyze

You are to find two quantities, using whatever computation is necessary:

_____ in Horizontal displacement of the actuation point

_____ in The largest input displacement that does not cause yielding (take yield stress to be 40 KSI)

When you are done, check your work, notify your instructor, and proceed without further delay. (over)

Step 3: Optimize

You are to meet two specified objectives, ± 10%, by changing the design parameters appropriately. Briefly consider your optimization strategy before you begin. You may **not** use the optimization tool.

Design parameters:

a) the depth of all beams (they must remain the SAME!)

b) the width of the vertical beams (they must remain the SAME!)

Objectives:

1) The structure fails when actuated 0.035", ± 10%

2) The structure displaces 0.035" under a 100-lbf load

When you are done, check your work, write what design parameters you chose, and notify your instructor.

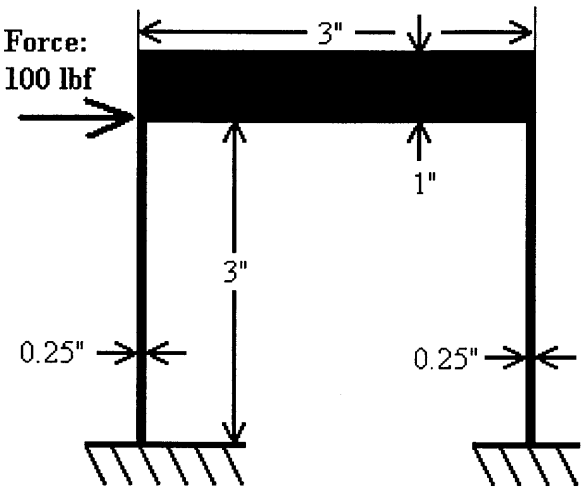
Optimized design parameters:

a) the depth of all beams _____ in

b) the width of the vertical beams _____ in

Compliant Mechanism Design Problem, Cosmos version

You are to create, analyze, and optimize the compliant mechanism below. Please read all instructions carefully. Good luck!



Depth: all beams 0.3"

Step 1: Create

Create the mechanism as shown. All beams are 3" long and 0.3" deep. The vertical beams are 0.25" wide, while the horizontal beam is 1" wide. A horizontal 100-pound force is applied as shown.

When you are done, check your work, notify your instructor, and proceed without further delay.

Step 2: Analyze

You are to find two quantities, using whatever computation is necessary:

_____ in Horizontal displacement of the actuation point

_____ in The largest input displacement that does not cause yielding (take yield stress to be 40 KSI)

When you are done, check your work, notify your instructor, and proceed without further delay. (over)

Step 3: Optimize

You are to meet two specified objectives, ± 10%, by changing the design parameters appropriately. Briefly consider your optimization strategy before you begin. You may **not** use the optimization tool.

Design parameters:

a) the depth of all beams (they must remain the SAME!)

b) the width of the vertical beams (they must remain the SAME!)

Objectives:

1) The structure fails when actuated 0.025", ± 10%

2) The structure displaces 0.025" under a 250-lbf load

When you are done, check your work, write what design parameters you chose, and notify your instructor.

Optimized design parameters:

a) the depth of all beams _____ in

b) the width of the vertical beams _____ in

Bibliography

[1] Bridge software links. www.bridgesite.com/softwarelinks.htm.

[2] Getting started with abaqus/standard. MIT class 2.094 lecture handout.

[3] Matlab function reference. http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml, 1994-2001. The MathWorks, Inc.

[4] G.K. Ananthasuresh, S. Kota, and Y. Gianchandani. A methodical approach to the design of compliant micromechanisms. *Solid-State Sensor and Actuator Workshop*, pages 189–192, 1994. Hilton Head Island, South Carolina.

[5] Gordon A. B. Anderson. A six degree of freedom flexural positioning stage. MSME thesis, Massachusetts Institute of Technology, Cambridge, 2003.

[6] Cascade Consulting Associates. StruCalc 5.0 demo tour. www.strucalc.com, April 2002.

[7] J. R. Barber. *Intermediate Mechanics of Materials.* McGraw Hill, Boston, 2001.

[8] Robert D. Cook. *Concepts and Applications of Finite Element Analysis.* John Wiley & Sons, New York, 4^{th} edition, 1974.

[9] Stephen Crandall, Norman Dahl, and Thomas Lardner. *An Introduction to the Mechanics of Solids: Second Edition with SI Units.* McGraw-Hill, New York, 2^{nd} edition, 1978.

[10] Martin L. Culpepper, Gordon A. B. Anderson, and Patrick A. Petri. Hexflex: A planar mechanism for six axis manipulation and alignment. *Proceedings of ASPE 17th Annual Meeting*, November 2002.

[11] Norman E. Dowling. *Mechanical Behavior of Materials.* Prentice-Hall, Upper Saddle River, New Jersey, 2^{nd} edition, 1998.

[12] M. Frecker, S. Kota, J. Fonseca, and N. Kikuchi. A systematic synthesis method for the design of distributed compliant mechanisms. *AMR '95 - The Fourth National Applied Mechanisms and Robotics Conference*, 2(062), December 1995. Cincinnati, OH.

[13] Kurtis R. Gurley. Direct stiffness: Beam application. CES 4141 Lecture Notes, Summer 1998. www.ufla.org/kgurl/.

[14] Layton C. Hale and Alexander H. Slocum. Optimal design techniques for kinematic couplings. *Precision Engineering*, 25, 2001.

[15] O. Sigmund. On the design of compliant mechanisms using topology optimization. *Mechanics of Structures and Machines*, 25(4):495–526, November 1997.

[16] Kenneth J. Waldron and Gary L. Kinzel. *Kinematics, Dynamics, and Design of Machinery.* John Wiley & Sons, New York, 1999.

[17] Warren C. Young and Richard G. Budynas. *Roark's Formulas for Stress and Strain.* McGraw-Hill, New York, 7^{th} edition, 2002.